Faculty of Computer Science
And
Information Technology
University of Malaya

Bachelor of Science Computer

Interactive 3D Driving Test Simulation (I3DDTS)

Syahizad Zuriatie Bt. Mohamad Zuki
WEK010267
Department of Computer Network and Technology

Supervisor: En Amirrudin B. Hj Kamsin
Moderator: Cik Mas Idayu Bt. Mohd Sabri

ABSTRACT

Interactive 3D Driving Test Simulation is a project develops a virtual world of interactive 3D with simulation. There are some major features associated with the system. This project system is not only provides better services for the user but also helps immensely to the driving school internal management.

The purpose of this project was to identify innovative and feasible applications of existing simulation and other electronic device technology with clear potential for the safety training of novice drivers and professional users through an effective lesson using this courseware.

This Interactive 3D Driving Test Simulation project is divided into two main parts which is to create the model of the component that is included in the driving test and integrate all the models to form a fully function system.

The model that are included in this Interactive 3D Driving Test Simulation are the driving test environment itself such as the road, housing, roundabout, traffic light and so forth. The most important model is the vehicle that is use by the user in the 3D driving test. The tools that are going to be use to develop the model for these projects are three-dimensional software.

The engine development is the major part to integrate the entire model to form a fully function system. A driving test simulation is a virtual reality tool that makes the driver feels like driving in a real car. A driver simulator normally consists of several subsystems such as real-time vehicle dynamics simulation, motion and force feedback, virtual reality system, rendering graphics and transferring the simulation data and more.

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to the following persons without whom this project would never been completed.

First of all, my foremost gratitude goes to my respected supervisor, **En Amirrudin Haji Kamsin** (Lecturer of Faculty Science Computer and information technology) for devoting his precious time providing the generous guidance, support, dedication, encouragement, supervision, and constructive advice throughout the whole development stage of Interactive 3D Driving Test Simulation with patience and dedication. Being an undertaking and open-minded lecturer, he has provided a platform for student, especially my self to explore and work independently without any fear of wrongdoing.

Especially thanks to **Pn. Nornazlita Hussin** (Lecturer of Faculty Science Computer and information technology), the project moderator who contributed suggestions, comments, and ideas to further enhance value of this project. The appreciation is also dedicated the others who have provided important information for the project development. Not forgetting, document room, all FSKTM lab administrators who have made sure that all computers and other facilities in the labs are in good condition.

Special word of thanks to my partner, Corina Binti Jeffery Minggu, who helped me a lot in many ways especially during the completion of this project.

To all my housemate, course mate and friends, Tengku Azlina Tengku Ariff, Noor Fadilah Khalil, Nusrah Ghani, Haniza Yazid, Sh. Haslinda Sy. Boniamin, Sh. Syafinaz Sy. Abu Bakar and many others, thank you for rendering your friendship and at the same time for being supportive and encouraging all these years.

My deepest appreciation to Puan Zalinah Ithnin, clerk of Surfine high-tech Driving School and En Fatah, manager of Sekolah Memandu Semasa, for their contribution towards the preparation and analysis of this project.

Last but not least, special thanks to all my family members especially my parents En Mohamad Zuki and Pn Shahmah Hamid, who had been very loving and encouraging all these years.

Thank you very much …

TABLE OF CONTENT

LIST OF TABLES

CHAPTER 1

INTRODUCTION

CHAPTER 2

LITERATURE REVIEW

CHAPTER 4

SYSTEM ANALYSIS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

CHAPTER 2

LITERATURE REVIEW

CHAPTER 3

PROJECT METHODOLOGI AND REQUIREMENTS

CHAPTER 5

SYSTEM DESIGN

CHAPTER 1

INTRODUCTION

Advances in computer technology have made interactive 3D graphics systems a common tool for a variety of areas. Most of such systems rely on popular 2D devices. To relieve system developers from the burden of implementing mapping techniques between 2D devices and 3D application data, some generic libraries have been proposed. However, those libraries still present limitations, either being complex to program or being limited in its interactive flexibility.

This project is taken up with the intention to improve the user skills and familiarize them with traffic law and continuous analysis of traffic law in Malaysia in order to schedule and replenish needed to satisfy anticipated demand. Besides, it's also to serve the purpose of maximizing profit with a clear focus on improved efficiency and effectiveness.

Just like in a real driving situation on the road, the driver here is also integrated in the driving process as a controller via the visual feedback of the driver. For standard applications in exhaust emission certification as well as for new areas of application such as gear and brake optimization, this project has been adapted to the test task and the test driver to the best possible extent with a view to ergonomics and space perception. Thus the driver can carry out his tasks over long periods of time in a concentrated and at the same time relaxed way and with a high degree of work satisfaction. This project was extended by important operating and visualization functionalities but also contains all standard functionalities and interfaces.

1.1 PROJECT OVERVIEW

The increases of road accident in Malaysia have become one of the most anxiety matters in our country. Almost everyday there is a news about accident in the newspaper or other electronic media. Below is the statistic from the Royal Malaysian Police (Polis Di Raja Malaysia) for the road accidents statistic in Malaysia [1].

Accidents

Accidents	**1997**	**1998**	**1999**	**2000**	**2001**	**2002**
Death	5,703	5,190	5,219	5,440	5,230	5,372
Serious	12,158	10,129	8,536	8,067	6,942	6,706
Less serious	32,204	33,071	31,241	28,778	30,684	30,319
Damage	165,567	162,644	178,170	208,144	222,319	236,840
Total	215,632	211,034	223,166	250,429	265,175	279,237

Road User

Road user	**Injuries**	**1997**	**1998**	**1999**	**2000**	**2001**	**2002**
Walker	Death	763	694	707	721	645	648
	Serious	1496	1245	1042	914	802	764
	Less serious	2795	2849	2817	2260	2274	2450
Drivers	Injuries	1997	1998	1999	2000	2001	2002
Motorcycles	Death	3286	2981	2960	3118	2971	3030
	Serious	8385	6978	5804	5369	4576	4483
	Less serious	22922	24577	23474	21622	22801	21,635
Bicycles	Death	251	249	291	236	232	235
	Serious	732	612	532	448	403	345
	Less serious	1538	1708	1634	1321	1430	1391
Motorcars	Death	447	404	421	543	527	559
	Serious	684	642	590	665	669	656
	Less serious	2955	2442	2677	2830	3139	3227
Vans	Death	24	77	62	64	65	70
	Serious	73	134	106	117	88	105
	Less serious	195	443	410	345	343	351

Bus	Death	21	11	6	10	8	17
	Serious	40	26	22	32	27	18
	Less serious	119	70	82	63	85	91
Lorry	Death	196	112	119	114	107	129
	Serious	210	149	127	137	125	135
	Less serious	654	524	464	489	435	549
4-wheel drive	Death	21	40	31	39	51	32
	Serious	20	41	49	45	39	43
	Less serious	67	142	152	292	198	249
Others	Death	196	49	52	47	48	46
	Serious	307	86	70	72	92	74
	Less serious	813	274	263	255	302	285
Passengers	injuries	1997	1998	1999	2000	2001	2002
Motorcycles	Death	474	428	450	401	398	397
	Serious	1215	1128	1043	953	816	808
	Less serious	2035	2356	2369	2188	2065	2255
Bicycles	Death	4	18	18	11	12	33
	Serious	24	43	54	32	30	48
	Less serious	47	95	110	80	101	123
Motorcars	Death	432	404	427	487	517	456
	Serious	591	523	515	603	638	628
	Less serious	1249	1294	1339	1592	1741	1484
Vans	Death	44	83	75	86	84	85
	Serious	118	189	147	173	127	127
	Less serious	187	369	326	371	353	337
Bus	Death	54	40	35	37	60	29
	Serious	101	115	96	86	104	65
	Less serious	292	370	285	302	344	303
Lorry	Death	38	79	66	66	68	69
	Serious	42	86	78	70	68	61
	Less serious	120	167	155	189	118	195
4-wheel drive	Death	19	51	46	34	40	41
	Serious	13	41	53	46	40	50
	Less serious	41	108	127	99	119	148

Others	Death	32	20	28	21	16	11
	Serious	54	30	38	28	36	14
	Less serious	138	108	93	77	96	98

Table 1.1: Statistic of Road Accident in Malaysia

In Malaysia, the number of road accidents keeps increasing every year. Road accidents statistic shows that 250, 429 cases occurred in year 2000. Whilst in year 2001, the amount increases to the total of 265,175 cases. This number keeps increasing in the year 2002, with total of 279,237 cases. From this statistic, the most involved vehicles recorded were motorcar. Here alone, we can assume that motorcar drivers are facing the highest risk of accident in its range.

The purpose of this project was to identify innovative and feasible applications of existing simulation and other electronic device technology with clear potential for the safety training of novice drivers and professional users.

The project that I am about to build is divided into two main parts. The first part is to create the model of the component that is included in the Driving test. The second part is to integrate all the models to form a fully function system. My partner Corina binti Jeffery Minggu will be doing the first part of the project. While I will be cover the second part of the project.

The model that are included in this Interactive 3D Driving Test Simulation are the driving test environment itself such as the road, building, trees, traffic light and so forth. The most important model is the vehicle that is use by the user in the 3D driving test. The tools that are going to be use to develop the model for these projects are three-dimensional software.

The engine development is the major part to integrate the entire model to form a fully function system. A driving test simulation is a virtual reality tool that makes the driver feels like driving in a real car. The virtual traffic environment gives the driver impression that he or she is driving in the real world. A driver simulator normally consists of several subsystems as follows:

- Real-time vehicle dynamics simulation
- Motion and force feedback
- High resolution visual and audio feedback
- Virtual reality system
- Rendering graphics
- Transferring the simulation data.

1.2 PROJECT DEFINITION

Interactive 3D Driving Test Simulation is a project develops a virtual world of interactive 3D with simulation. There are some major features associated with the system. This project system is not only provides better services for the user but also helps immensely to the driving school internal management as describe below:

1. ***For user:***
   - Familiarize the user with driving test environment before they undergo the real test.
   - Provide user with close situation of driving test surrounding.
   - Sharpen the driving skills of user virtually
   - Enable user to gain more information on traffic laws.

2. ***For management:***
   - Prepare the management staff with the software.
   - Provide a simple user friendly system to obtain more information about driving test simulation.
   - Design to the need of the real driving test purposes to driving school.

1.3 STATEMENT OF PROBLEMS

Problems of Current Driver training

Driving test is one critical process of giving one a license to drive. Efficient driver education will produce graduated driver, which will not only pass the driving test, but at the same time is a safe driver on the road. With the help of science computer technology, we can create a program that will help in driver education. It is a courseware that gives student a change to practice driving in a virtual environment and effectively teaches them to drive safe. Conversely, driver education without the aid of computer technology or the traditional driver education methods most likely to face the following problems:

1. ***Traditional driver education methods is costly***

   Existing driver education uses an automobile for the driving lessons. Therefore, driving schools have to spend huge sum of money for vehicles maintenance as well as the insurance and fuel. The more driving lesson they have, the more expenses they have to bear.

2. ***Weather Hazard***

   Weather is a phenomenon that we cannot control. When it is raining day, it would be dangerous to continue driving lessons as novice driver posses very few knowledge on driving. The slippery road which is dangerous for new drivers to drive often be the factor for driving lesson being canceled due to no other alternative training.

3. ***Time consuming***

When the lesson session, Student often wasting their driving range waiting for their turn while another student is in the vehicle. So, this software can be other alternative for student to practice rather than wasting their time without doing anything.

4. ***Driving Instructor often loose their temper during driving lessons***

Not every instructor is passion while teaching their student. In various cases, instructor often looses their temper and shouted on students. This kind of instructor is almost certain not able to teach their students effectively.

5. ***Minimal training period per week***

If the driving school is that of the popular one, it is certain it has a large intake of student in their school. Each student driving lessons period per week will be very minimal in order to cope with the large amount of student to be educating in their school. Therefore, training period will be longer and student has to pay more for the fees before they can undergo the driving test.

1.4 PROJECT OBJECTIVES

- Create an interactive virtual environment for driving test using steering controller, mouse or keyboard.
- Interactive 3D driving test simulation includes lessons with the route that have been approved by the Ministry of Transport, so users can customize and trained virtually before the real test.
- Identify the deficits in knowledge, skills, or attitudes among novice drivers that are most amenable to enhancement through simulation, and the user interface (display and control) requirements to convey program content to novice drivers.
- Determine the most-cost-effective, accessible devices to meet the program content and user interface requirements.
- Identify techniques which will produce reliable and desirable peer influences on safe driving behavior.
- Provide a simple user-friendly system to obtain more information about driving test simulation.

Target User

Target user consists of two categories which are:

- Professional road user

  Professional road users are consisting of approved instructor and traffic officer.

- Public

  For the public user especially for novice user and those who are going to undergo a driving test.

1.5 PROJECT SCOPE

The scopes of the project are divided into three parts which are Limitation, Assumptions and Constraints.

- **Limitation**

  The scope of this project covers the development of 3D Driving test in three-dimensional visualization. The project will be developed on windows platform. The project covers all topics from defensive driving, road rage, and even the new graduated licensing requirements. The scope of topics to be taught in the package is limited to 3D topics in computer graphic.

- **Assumptions**

  The assumption for the project is majority of system's users will be from targeted users of the system. The final product will be stored on a CD and can be used on any stand-alone computer with a CD-ROM drive.

- **Constraint**

  The boundaries of the project development are:

  1. The consideration for the complexity of system would be highly based on the availability of the technology and affordability of hardware or software acquisitions.
  2. The project need a conscientious analysis and requires continuous effort to finish the project.

Project Boundaries

- ***General Abilities Driving Elements***

  General abilities driving elements are fundamental visual/perceptual information processing skills with potential benefits for a wide range of specific driving skills. When driving general abilities, it is assumed that skills developed will generalize to a wide range of driving tasks. As an example, training that emphasizes scanning the environment for potential hazards would have benefits for multiple driving tasks such as lane changing, speed choice, and left turn decisions. [31]

- Examine situational awareness issues early in development.

1.6 EXPECTED OUTCOME

The Interactive 3D Driving Test Simulation is expected to achieve the following outcome:

- System can perform some basic function and meet some importance criteria such as stability, consistency, reliability and user friendly.
- A system that will be able to adapt in a real life driving test surrounding into a 3D image.
- An interactive, pc-based application manages to deliver effective, flexible and efficient drivers education.
- System with a user friendly interface to enable user undergo training in an exciting environment.
- A system that can be categorized as a courseware that provides interactive virtual drivers training.
- A system that able to function using a steering, keyboard and mouse control.
- A system that will be able to adapt in local driving school environment.
- The final implementation should allow for future enhancement as well as additional modules to extend the system functionality.

1.7 PROJECT SCHEDULE

Several form of planning was made to ensure the smoothness of project development. The following show the System development chart:

Semester	Major Activities	Minor Activities
Semester I	• Literature Review • Requirement Analysis • System Design	• Data gathering • Data flow and Decision Analysis • Proposal Preparation • Specify System requirement • Specify System design • Submission of Report(WXES 3181)

Table 1.2: Five major activities in project planning

The most important phase is the ***data gathering activity***. Data are gathered from primary (user), secondary (parties whom recites based on primary sources) and tertiary (magazines, book, internet, newspaper, journals and so forth). Below are details activities of Analysis Planning and Implementation Planning.

Activities	Detailed Activities	Duration(Weeks)
Data Gathering	• Preliminary investigation on the topic. • Survey and interviewing. • Read relevant information resources. • Assemble all data collection	4
Data flow and Decision Analysis	• Analyze data • Introduce user Interface Design Prototype	4
Proposal Preparation	• Prepare proposal • Present proposal	2

Table 1.3: Detail activities of Analysis Planning

Activities	Detailed Activities	Duration (Weeks)
Implementation	• Assemble all data ✓ Assemble all specific data about driving test. ✓ List out all the images and models to build. • Integrate all the models to form fully system. • Create Interface ✓ Create package interactive as interface. ✓ Import 3D models into package ✓ Building links and navigation page	14
Evaluation	• System Testing ✓ Error detection ✓ Fixing the error • Final Evaluation • User training • Documentation	4 Continuous

Table 1.4: Detail activities of Implementation Planning

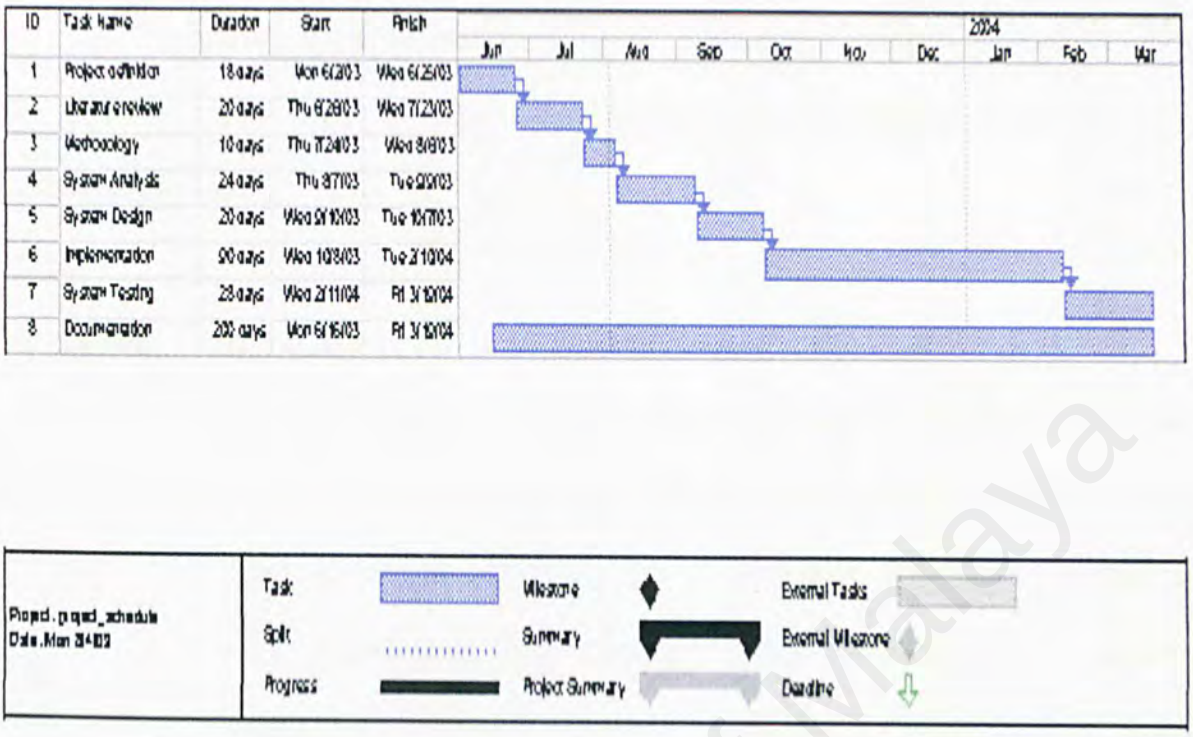


Figure 1.1: Project Schedule of Interactive 3D Driving Test Simulation

1.8 REPORT LAYOUT

This project proposal report consists of eight chapters. The purpose of this layout is to give overview of the major phases involved during development of the project. Below is the report layout:

Chapter 1: Introduction

This chapter gives an overview of the major phases of the project that includes the introduction, project overview and project definition, project objectives which state purpose of the system plan to be done before project scope, project significance, system strength, project resources, project schedule, and summary. It includes the expected outcome of the system, project scope, and the limitations of the system.

Chapter 2: Literature Review

This chapter gives brief explanation on topics researched and studies that are relevant to this project. It is the combination between literature search and literature review. Among the discuss topics are development tools and technology including operating system, system architecture and programming languages. It is a very important stage in system development. It is an approach to search and gather all the system requirements and related information to develop a new system. It is done through some finding on related books and senior's thesis, articles available from the Internet and so

on. In this section, searching, summarize, analysis and synthesis of the system are done.

Chapter 3: Methodology

This chapter emphasizes on the justifications for the chosen project methodology. It also discusses the information gathering techniques and the explanation about the development software and platform chosen to develop this system. It clearly identifies the methodology, mechanism, and approach to be adopted in developing this project system. It extensively defines the project development methodology on system development, capturing information and analyzing system requirements. All users' requirements and administrative software and hardware are listed below. It also discusses the information gathering techniques and the explanation about the development software and platform chosen to develop this system.

Chapter 4: System Analysis

This chapter describes the system analysis of the project including functional requirements, non-functional requirements, hardware and software requirements on different developing tools. It also explains how the requirements for this project were acquired.

Chapter 5: System Design

This chapter explains the conceptual and technical design of the system. It covers the structure chart, data flow diagram, process flowchart, and user interface. System designs are focuses on the architecture design of this Interactive 3D driving Test simulation system, and process/functional design as well as the interface design of this project. It includes the tables and the diagrams that are used to depict a pictorial idea of this system development. It highlights the designing program, expected outcome, user interfaces and covers structure chart, data flow diagram and process flowchart.

Chapter 6: System Implementation

This chapter consists of the detail explanation of the implementation phase and the coding process involves transforming of the design into a programming language.

Chapter 7: System Testing

This chapter will discuss about the testing phase. This is also a very important stage whereby testing is essential to assure quality of the system. The objective of testing is to find system error and fault.

Chapter 8: System Evaluation

This chapter will see the system evaluation. System evaluation will touches various things like problems encountered during the development process, system strength and limitation and others.

1.9 SUMMARY

This chapter focuses mainly on the introduction of this project. A brief introduction and definition are stated in the first part of this chapter, which is the Project Overview. Apart from that, relevant information and topics are also being discussed consequentially. Topics included are Project Definition, Project Objectives, Project Scope, Expected Outcome, Project Schedule, and Report Layout. The research and development of this proposed system will take about 8 months.

The next chapter literature review will carried out whereby current systems are surveyed to better understand how it is implemented, together with comparisons between different operating system platforms, development tools and others.

CHAPTER 2

LITERATURE REVIEW

2.1 DOMAIN STUDIES

This subsection contains all the research has been done on the existing current system including reviews on the features, capabilities, and so on. The weaknesses on existing current system were identified in order for this project to overcome and the strengths of existing interactive 3D driving test simulation were studied so that it can be adapted into this project.

2.11 DEFINITION

Interactive

Describe a system or a mode of working in which there is a response to operator instructions as they are input. The instruction maybe presented via an input device such as a keyboard or mouse, and the effect is observable sufficiently rapidly that the operator can work almost continuously. The mode of working is thus sometimes referred to as *conversational mode.* An interactive system for multiple users will achieve the effect by time sharing. [2]

3D stands for Three-Dimensional. In computers, 3-D (three dimensions or three-dimensional) describes an image that provides the perception of depth. When 3-D images are made interactive so that users feel involved with the scene, the experience is called **virtual reality**. 3D always refers as **3D graphic** or **3D animation**

Driving

The act of controlling and steering the movement of a vehicle or animal. [13]

Driving Test

Test that must be passed to obtain a driving-license. [27]

Simulation

The scene comprises objects moving around and interacting with each other and the rest of the virtual world, and we attempt to create a reasonably realistic experience for the user. [2]

Virtual Environment

Virtual Environment (Kooper 93): A computer generated world with which the user can interact. Interaction can vary from looking around to interactively modifying the world. [26]

Rendering

The part of computer graphics that is concerned with getting from a three-dimensional scene (possibly containing moving objects) to a picture or animated sequence, with more or less sophistication in terms of the effect achieved. [2]

2.12 ACRONYMS AND TERMINOLOGY

Virtual Environment

A Virtual Environment [31] is defined as a system which immerses the user in a computer-generated environment using, in addition to high-speed computer graphics technologies, head-mounted displays (HMDs) and sensors that detect body movement, such as head trackers and gloves that detect finger movements (Levine and Mourant, 1995). There are technological problems with the current batch of Virtual Environment devices. First, the graphics are orders of magnitude poorer than what can be obtained on the other platforms. Furthermore, these VE systems slow update rates for the graphics, which significantly limits the applicability of the device as a training platform. These shortcomings can be readily overcome using faster CPUs and graphics accelerators that cost much more. In terms of using Virtual Environment technology for novice driver training purposes, it is possible to discuss the potential benefits of future low-cost Virtual Environment devices. In terms of technological capability, it is conceivable that Virtual Environment will one day be entirely immersive, that is the visual information received from the Virtual Environment will eventually be indistinguishable from behind-the-wheel driving. However, the effects of user inputs to the Virtual Environment (i.e., steering, braking, and accelerator control reposes of the user) will never be rendered instantaneously, since, there is a finite amount of time required to recomputed a graphics image based on a user's input. Whether this lag will be undetectable as computer processing speed increases is unknown. The amount of Virtual Environment lag is also influenced by the total number of pixels processed in the image, which relates to the field-of-view versus resolution tradeoff. Virtual Environment technology

also has the potential benefit of allowing multiple users to interact within the Virtual Environment. Instead of the trainee responding to events that originated in the computer software, the trainee responds to other users whose responses are unpredictable, similar to actual on-road driving (Decina et al., 1996). There are several types of Virtual Environment hardware currently on the market, including head mounted displays, glasses, gloves, joysticks, and vests.

Interactivity

For the purposes of this project, interactivity [31] is defined as the quantity and quality of auditory/visual feedback that correlates with user responses so as to simulate behind-the-wheel driving. In order to be considered an interactive simulation, a training program must have at least one response that satisfies this definition. At the low fidelity end of the interactive response continuum, a simple button press can be used to simulate an interactive brake response (in this case, the button press freezes the scene). Since all platforms can implement this level of response fidelity, it is not anticipated that this will distinguish among targets. Instead, somewhat higher response fidelity will be used to distinguish among platforms. For example, a higher fidelity braking response would allow the driver to control the deceleration in proportion to the pressure applied to a brake pedal. A somewhat more difficult interactive response to implement is a steering task. In most cases, implementing a high-quality steering response requires the use of computer graphics images (as is typical in video arcade games).

The following two interactive response categories are discussed in the evaluation of potential training platforms:

1. Simulated vehicle speed control (required to simulate brake and accelerator responses).
2. Simulated vehicle direction control (required to simulate navigation tasks, turning, lane changing and other tasks that require steering wheel inputs). The training requirements for these two types of responses can be determined from the scripts described previously.

Assessing the interactive response capability of platforms depends on whether computer graphics or actual video are used as stimuli. As a general rule, the ability to implement interactivity in a low-cost training platform, including development costs, requires the use of computer graphics images. Although creative script development can reduce the costs when using actual road scene video, this approach is not recommended. Thus, if the use of actual video in the training is proposed, it is unlikely that the platform can be used to implement either of the interactive response categories. If computer graphics are proposed, the capability for implementing higher quality or more response categories is more likely.

2D and 3D animation

The field of research and development new demands arises such as the optimization of gear and brakes on the roller. For this purpose among other things the simulation of hill and dale driving or starting on a slope is necessary. A realistic display of the road situation is absolutely necessary. Farsighted driving shall be possible for the driving test just like during real road driving. This makes high demands on the man-machine interface.

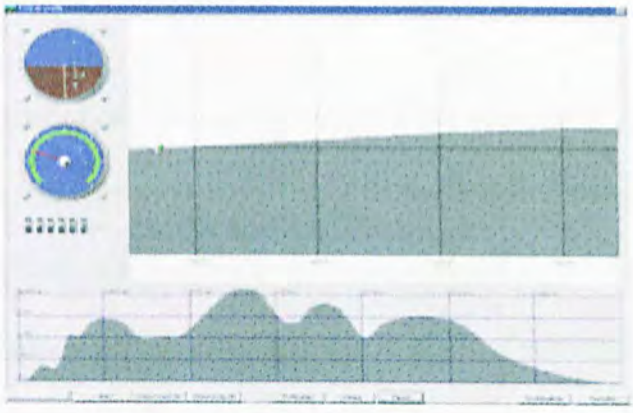


Figure 2.11: Example of 2D-animation overall view of the test section **[29]**

3D-animation [29] presents a road drive into a virtual picture. Bare landscape which reproduces the slope and bend situation. In order to be able to edit significant points freely in this bare landscape. Objects like demarcation poles, street signs, traffic lights, bus stops, forests, cities, bridges, highways, intersections, petrol stations, fire hydrants, telephone boxes etc. at any given point in the bare landscape are viewed in interactive way.

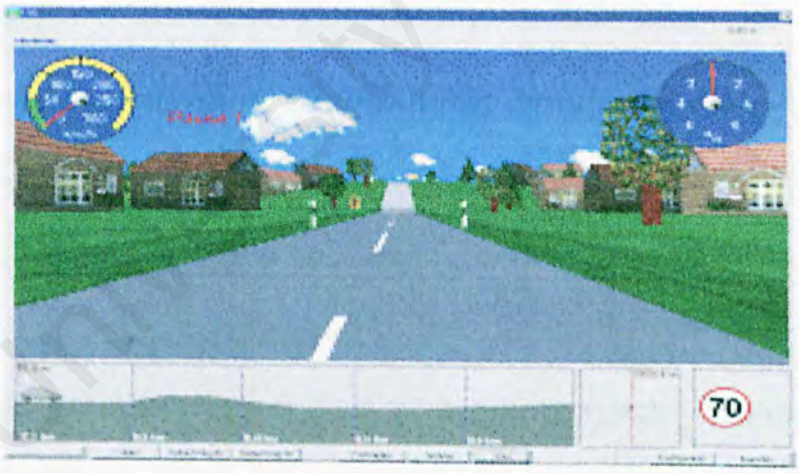


Figure 2.12: Example 3D-animation with GPS-interface and comfortable landscape editor **[29]**

Driving Test

Driving is the act of controlling and steering the movement of a vehicle or animal. In this project, driving is certainly referred to act of controlling and steering the movement of a vehicle. **Driving Test** is an examination to determine whether a driver has gained competence enough to drive on the road. The test consists of two examinations - a theoretical test and a practical driving test. The test taker has to pass the theoretical test before he or she is allowed to take the driving test.

Simulation

A computer model of a real phenomenon or system. A 3D simulation [30] is described by 3D models in a computer program. Simulations are used in computer games, training programs (flight simulators) and by scientists, who recreate, project into the future and predict real world phenomena.

Engine development

The development of engine is first guided by the most fundamental decision whether it should become an entirely general engine that can be subsequently be used to develop almost every kind of engine such as game engine, or should be focused on a single genre, or maybe developed for only a single engine.

The Half-life engine uses a kind of hybrid approach, extending the inherently procedural Quake II engine with object-oriented game code written in C++. For the Parsec engine, a procedural programming paradigm was chosen and a C-like subset of C++, which is almost identical to ANSI C, was chosen as programming language. However, its architecture uses many object oriented concepts. (Hadwiger, 1990)

2.13 EXISTING SYSTEM REVIEW

Virtual driving in term of game has existed long enough that it became the number one in its range. However it is more to entertainment rather than educational purposes. The awareness on significant of safety driving has inspired developers to create a virtual driving that give drivers education with totally accident-free guarantee. Although it is a new approach in drivers education, there are already a few organization in western countries has implementing it. Later, a few systems that have some parts and concepts that similar to this project system will be discus.

CASE STUDY 1

Sekolah Memandu Surfine Hightech Sdn. Bhd. (Kampung Kerinchi branch)

- Currently they don't have any virtual driving system to assist in their driver education process.
- They have only two qualified driving instructor.
- They still keep to the traditional method of driver education which most of driving school in Malaysia does. Their drivers education usually only emphasized on the on road practice.
- The first-timer that registered to their company will advise to buy an L-guide book that cost RM 5.00. Then one has to study the book by them self in order to pass the theoretical test which consist of 50 objective questions.
- The on road practice schedule is fixing according to the instructor's timetable and user's convenient. Any weather hazard on the practice day will cause the practice to be canceled although you have reached the destination.

- During the driving lesson, you will have to wait without doing anything on the drivers range while someone is in the car.

CASE STUDY 2

Sekolah Memandu Kereta Semasa (Jalan PantaiBaru, Kuala Lumpur)

- Currently they don't have any virtual driving system to assist in their driver education process.
- They have three qualified driving instructor.
- They also using traditional method for driver education.
- Usually the practice schedule is fix base on working or non-working person where for working person the training period will be held on weekend, while for non-working person the training will be held on weekdays.
- Any weather hazard on the practice day will cause the practice to be canceled although trainee has reached the training center.
- During the driving lesson, trainee will have to wait without doing anything on the drivers range while someone is in the car.

CASE STUDY 3

Driving In the Virtual World [3]

(http://www.vrac.iastate.edu/~fang414/Publications/Paper_Driving.doc)

The Driver Simulator, the idea coming from flight simulators, has begun to appear in primitive forms in the 1970s. The Driver Simulator has been noted as an efficient tool for vehicle development, human factor study, other purposes since it is safe, experiment-repeatable and tightly environment controllable. Since then, many automotive makers and research institutions have developed and applied their own simulators to different research areas.

Graphics and Virtual Reality (VR) technology has made substantial progress on the construction for virtual traffic environment. VR first appeared in 1950s has became a popular research area since 1970s. VR has been applied to many different areas, such as Art, CAD, and medicine. VR brought a new era for computer graphics. Interactions with an immersive environment are a new evolution for 3D world.

System configuration

The simulator uses several computers to control the whole system and generate the graphics. One of computers is using Qnx operating system to simulate vehicle dynamics models and control the motion base and steering wheel. The C6 and C2 Virtual Reality system in Iowa State University are used to generate graphics and update traffic environment. The observer in the system can observe the whole traffic information and driver behaviors while a driver is driving in the virtual environment.

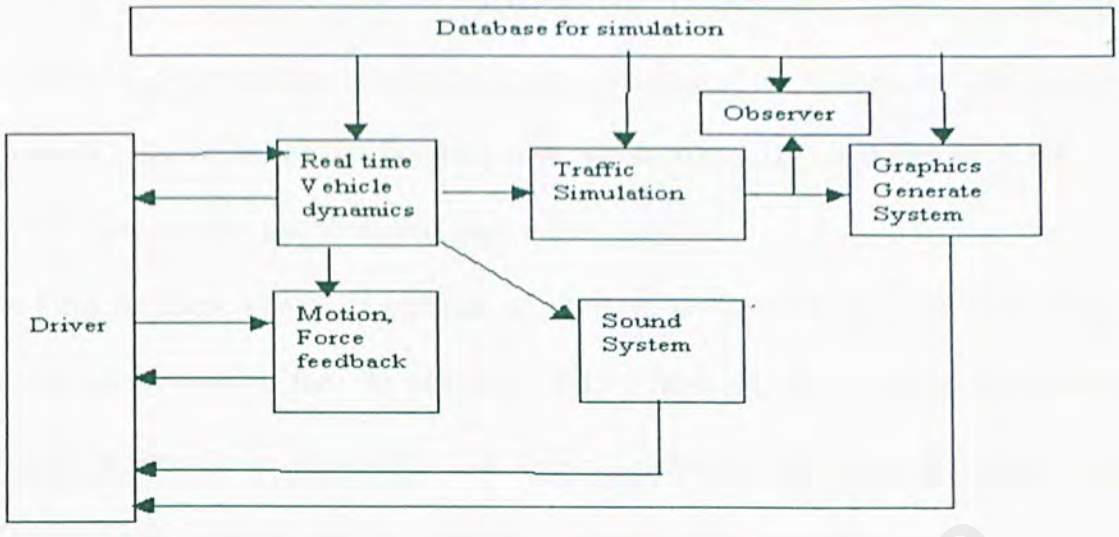


Figure 2.13: Driver simulator's system configuration

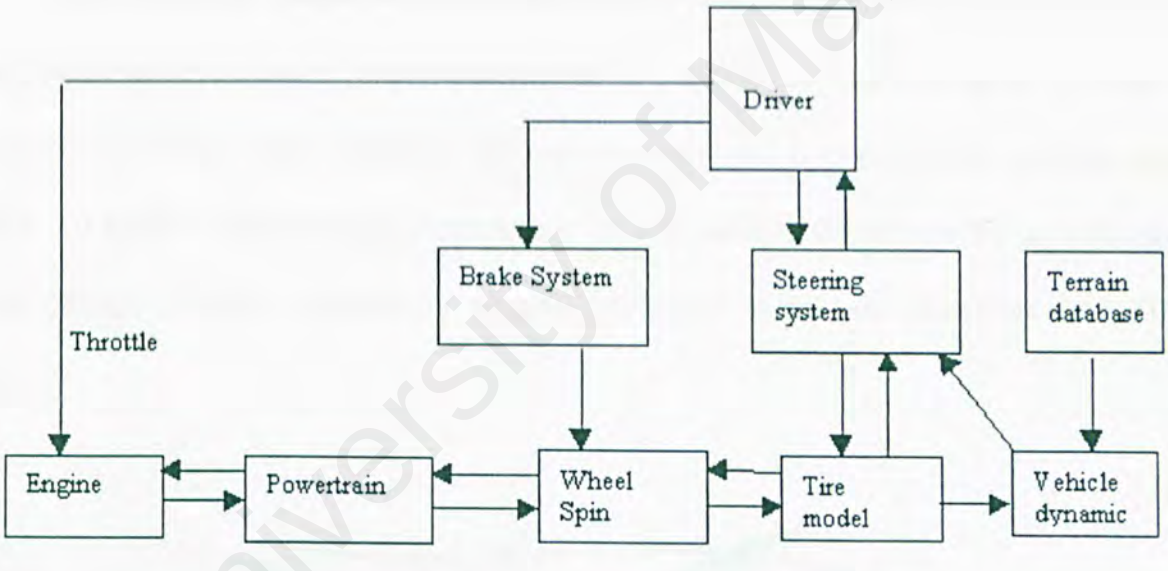


Figure 2.14: Real Time Vehicle Models

Real time vehicle simulation is a key element in driver simulators. In some car racing games, a driver would feel he/she is driving on the ice because of the lack of real time vehicle dynamics. The vehicle dynamics models include tire models, power train models, suspension models, and Engine models.

An electric six degree of freedom motion system from Sarnicola Simulation Systems has been used for the driver simulator in this study. Because of the limitation of the motion base, filtering is required to convert the computed motion to the simulated motion. A typical and classical washout filter has been used to attain this goal.

A force feedback system is another key element in feedback cues. Most of driver simulators focus on steering feel. An adaptive steering feedback is not enough to simulate realistic steering feel. A DC servomotor has been implemented into steering system. The steering torque varies with car velocity, steering angle and terrain properties.

Virtual Traffic Environment

Since the driver controls and maneuvers the car mostly depending on what he/she is seeing, the visual cue is a very important element for a realistic driving simulation. To study human factors using driver simulator, the real scenarios and high-resolution graphics are needed. To build a virtual traffic environment, several subsystems are needed as follows: terrain database, traffic management system, intelligent agents and scenarios, and 3D graphics.

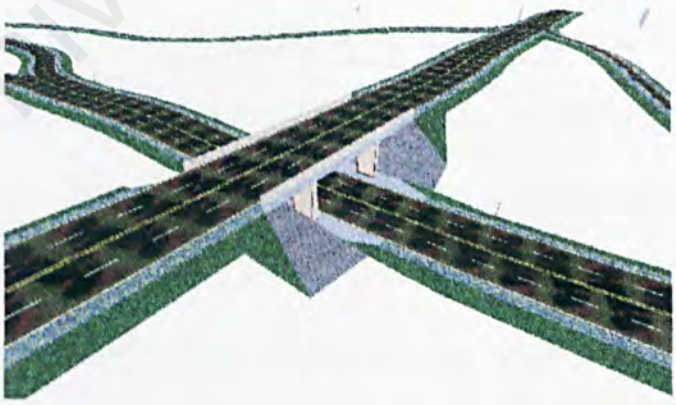


Figure 2.15: The road database in simulation mode

Terrain Database

The most key element in terrain database is the road database. The simulation is assumed each vehicle is driving on the road. If the driver drives a car off road, the simulation is over. Therefore, a simple terrain database for simulation only needs a road database. Other parts of terrain database are only useful for graphics. The two different databases are separated so that it is efficient for simulation and it is easy to add more graphics objects.

Driver simulators rely heavily on a driving simulation correlated database to provide the operator with a realistic virtual world. It is not only because the driver controls his/her car through visual cues, but also because the terrain information will highly affect the vehicle dynamics. Different terrain surfaces and grades will result in different responses of vehicles.

Traffic Management System

Traffic management system includes traffic lights and traffic signs. The traffic light system needs to update at each time step, while traffic signs are always the same through the whole simulation. The traffic system information is also included in terrain database. For an example, the lane speed limit is specified and stored when each lane is created in road editor.

Traffic light systems are different in different road junctions. Different traffic light systems have different control algorithms. When the junction is created in the road editor, the special traffic light system is needed to be selected. The control table, which controls each light in the light system based on the simulation time, is predefined. Therefore, when the simulation is running, the traffic light system updates its status depending on the simulation time has run. The control table could be dynamic, i.e. varies with the traffic density and daytime like in the real world. This would make the traffic management system more realistic.

Intelligent Agents and Scenarios

A lot of virtual traffic environments have been done for driver simulators. But few of them included intelligent scenes and scenarios. They are the hardest part in designing virtual traffic environments. It requires a well-designed database, driver behavior models, and artificial intelligence. A well-designed database would make cars get information about their surrounding environment and their own location easily so that the whole program is efficient. Otherwise, to design intelligent agents and scenarios is impossible. Driver behavior models are needed to study to make the traffic simulation realistic. Artificial intelligent is a higher-level driver behavior study. The fuzzy logic and neural network have been applied in driver behavior models.

3D Graphics

A high resolution graphics will make the virtual environment more realistic. Graphics database is different with the simulation database although graphics database is based on the simulation database. Graphics database is unlimited and has a lot of additional objects not including in simulation database.

Networking

Virtual reality system has more extensive applications with the network supporting. Two groups can collaborate and communicate through the network while watching the same virtual objects. It makes possible to solve problems on line and saves a lot of money on traveling. The network in driver simulators also has an important role. The multiple observers can sit outside the driver simulator, even in remote sides, and observe the behavior and the whole traffic simulation. In the application of network communication, a set of

simulation status data has to be sent to other sites. Considering the limitation of network rate, the data can not be so large.

CASE STUDY 4

Virtual Driving Simulator **[4]**

(URL: http://www.teendrivingcourse.com/DVDTraining/3dillustrations.cfm)

- The Virtual Driving Simulator is a revolutionary use of DVD that allows teen to practice driving in a virtual environment.
- Using the point-of-view of the driver, teen must navigate through busy streets, highways, and intersections and make real-time decisions.
- This section improves and trains driver awareness of surroundings and helps familiarize teens with what it actually looks like to be behind the wheel of a moving vehicle.
- **Practice Written Exam-**Using questions modeled after actual written driver exam tests, the practice exam will give teen a chance to get used to both the content and format of the written test.

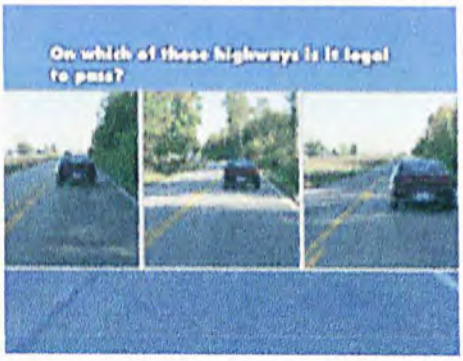


Figure 2.16: Interactive multiple choice questions reinforce the material presented in the driving lessons

- **The Driver's Exam-**In this section they follow their four teen instructors as they travel to the DMV, get behind the wheel, and take the test.
- **3D Illustrations**-They include 3D Illustrations that are available throughout the driving lessons when the "Show Me More" icon appears on the screen.
    - Parking at a 90 degree angle
    - The 3-second rule
    - Blind Spots
    - Four-way Intersection
    - High Beam Use
    - Parallel Parking
    - Parking on Hills
    - Passing on a Two Lane Highway
    - Dangerous Intersection.
    - When finished viewing, simply click "Return" to resume the lesson.

Figure 2.17: Each of these 3D Illustrations is viewable from a variety of angles by simply clicking the "Angle" button on DVD remote.

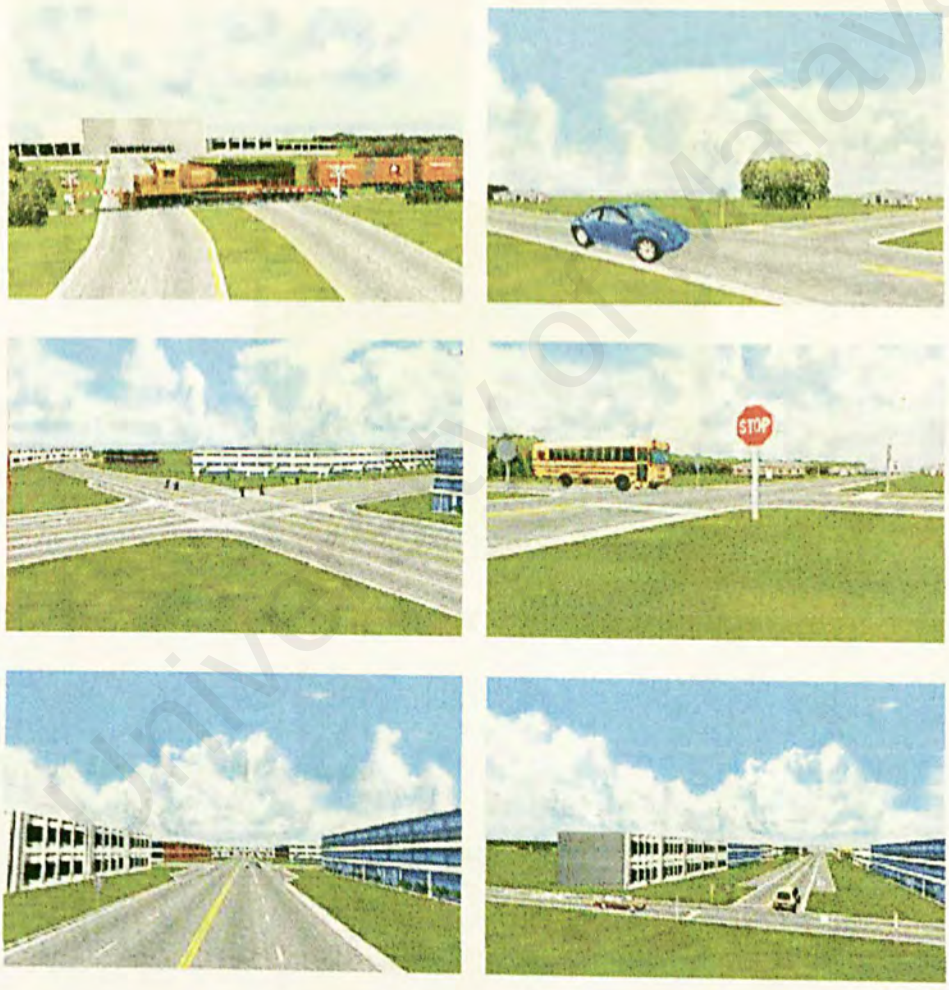
- **Lessons** -Each section is hosted by one or more of their four teen instructors. The instructors act as peers who provide helpful insight.
- **Learn by Doing**-They provide an interactive environment where the student can learn by doing. The teen instructors introduce situations such as parking on a hill with a downward grade and a curb and asking the student which way to turn their wheels. Then the student watches the outcome of their choice followed by an explanation from one of the teen instructors.
- **Quizzes**-At the end of every lesson they have included practice quizzes to test teen's knowledge of the information presented. In some situations, they illustrate via 3D Illustrations or a video segment the outcome of teen's decision.
- **Other Special Features**-They included material for driving in hazardous conditions, an entire section to Drug and Alcohol awareness as well as a message from the President of the Gateway Chapter of M.A.D.D. Safety tips also included such as checking your tire pressure, changing a tire, and checking your oil as well as some interesting, educational, and entertaining facts to keep the student interested.

CASE STUDY 5

Raydon Virtual Driver: Driving Simulators [5]

(URL: http://www.virtualdriver.net)

The Virtual Driver is a real simulator. Fully computer generated imagery and allows free movement and interaction within the virtual world. While immersed and driving through this virtual world, the students feel that they are actually there. It looks so real it motivates the students to learn more quickly and retain what they have learned.



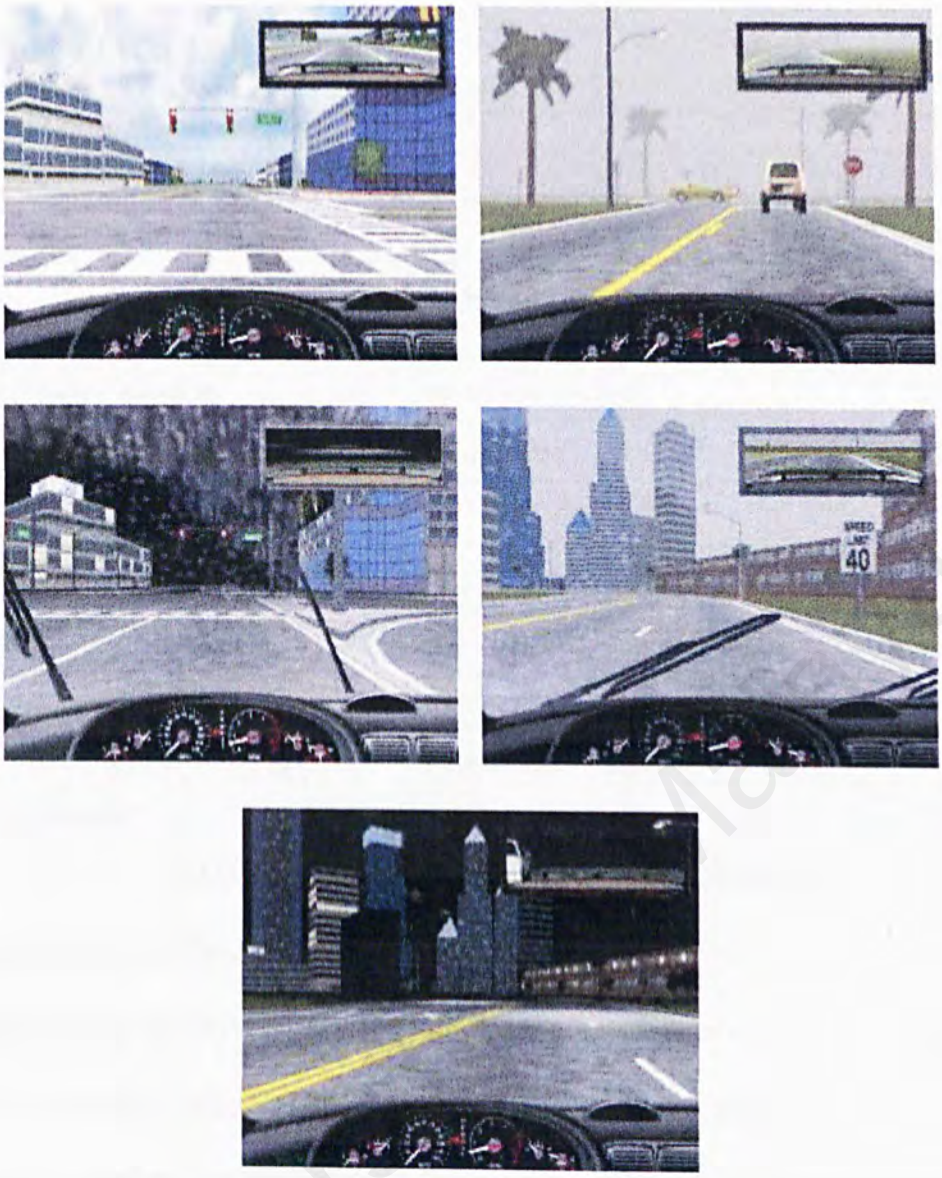


Figure 2.18: Virtual Driver Screen Shots

Raydon Virtual Driver Specifications

The Raydon Virtual Driver is compact, easy to maneuver and offers flexibility in classroom layout and design.

Frame Only	
Length	60 3/4"
Width	30 1/4"
Weight	205 lbs

Fully Assembled	
Length	60 3/4"
Width (keyboard stand stowed)	35"
Width (keyboard stand fully extended)	58"
Height (top of seat back)	48 1/2"
Height (top of monitor)	58"
Weight (including monitor and computer)	300 lbs
Operational Specifications	
Power requirements	120/240VAC 50/60Hz
Temperature	50° to 82°F (10°C to 28°C)
Relative Humidity (non-condensing)	20% - 90%"
Hardware Specifications	
Computer Monitor	19" Color VGA
Computer	Dual PentiumIII, 256 Megs of SDRAM, 20G Hard Drive, 52x CD Rom,
Software Specifications	
Operating System	Windows 2000 Professional
Visuals API	

Table 2.11: Raydon Virtual Driver Specifications

Driving Simulator for Driver Education

Raydon Virtual Driver - Driving Simulator offer users to explore how they can affordably add today's technology to improve driver education. Raydon's Road Tools multimedia curriculum and the Raydon Virtual Driver driving simulation system offer a comprehensive approach to teaching young drivers how to become competent and safe highway users. Together with Raydon's Performance Assessment System (PAS) Grade Book, the system provides an all-inclusive, fully interactive approach to driver education.

The Raydon Virtual Driver™ is the first fully interactive driving simulator system that puts students behind the wheel before they hit the highway. Our RVD™ driving simulators are a simple, cost-effective way to expertly prepare students for the challenges they will face on the open road. Cost-cutting measures continue to threaten even the most basic driver education classes at high schools across the United States.

Driving Simulator Training Concepts

The Raydon Virtual Driver driving (RVD) simulation system was designed by incorporating the same cutting-edge technology the company successfully honed through its development of military training simulators. Students learning to drive on the fully interactive Raydon Virtual Driver are benefiting from decades of training and simulation experience. The RVD simulator offers an active learning environment for today's driver education students.

The Raydon Virtual Driver is already an integral part of driver education classes. Students start with the Road Tools multimedia classroom package first. These two CDs are packed with three-dimensional animation, video, audio, text and tests. While the students are interacting with the curriculum and completing chapters, the instructor is able to monitor each student's progress from his own computer using the Performance Assessment System (PAS). This gives the instructor the ability to concentrate on each student's skill level and judgment development.

The Road Tools multimedia program and the Raydon Virtual Driver simulation system work together. As students progress through the preparatory lessons of the multimedia, they are then introduced to the "hands-on" driving simulation phase of the Raydon Virtual Driver. Students learn to apply the rules of the road, learn and practice proper driving behavior and skills and begin to apply risk assessment techniques. Because the RVD is fully interactive, it increases each student's actual "wheel time" and presents an environment that is safe, yet challenging for learning. After completion of this total training concept, your student will be better prepared to meet the challenges that a novice driver will face on the road.

Classroom setting Oak Ridge High School Orlando, Florida

Figure 2.19: Driving Simulator Training Concepts

About Raydon Virtual Driving Software

- The self-paced multimedia training incorporates 3-D animation, video, audio, text and applicable tests. Lessons are structured to allow students to progress at their own pace, giving them the opportunity to review information or get one-on-one instructional assistance from the teacher when they need it.
- The software is ideal for the high school environment. It operates on most PC-based systems, delivers extensive training value and requires minimal space and expense.
- Road Tools gives the instructor valuable feedback about the student's progress, enabling the teacher to work closely with each student on those segments of the program that might be more challenging than others. The system also offers incredible flexibility for students, teachers and administrators. Because their progress can be monitored by an instructor at any time with the Performance Assessment System (PAS), students can work on a lesson before or after class when their schedules allow.

CASE STUDY 6

Behavior Modeling and Scenario Authoring for Virtual Environments **[6]**

(URL: http://www.cs.uiowa.edu/~kearney/scenario.html)

This case study is a research to model agent behaviors and create replicable scenarios in large-scale virtual environments. Our goal is to reliably produce critical situations without restricting user's freedom of action, through inconspicuous direction of simulated agents. An important contribution of the project is the development of a modeling system based on hierarchical, concurrent state machines (HCSM) that supports the control of reactive behaviors and provides mechanisms for coordinating behaviors to create critical situations and events. A second research thrust is to develop new approaches for authoring virtual environments that integrate scene design (the layout of the environment) and scenario specification (what is to happen during a simulation). This technology has application in interactive, real-time simulation environments such as vehicle simulation for driver training, virtual prototyping, and experimental studies of driving safety.

Major features of a real-time driving and bicycling simulator:

- A behavior and scenario programming language and run-time execution environment based on HCSM.
- A real-time database system that represents and supports queries about topological, geometric, and logical attributes of the virtual environment, with special emphasis on accurate representation of 3D road networks. It also provides efficient proximity queries for dynamic entities, thus supporting real-time simulation of large numbers of vehicles and pedestrians.

- A real-time scheduler that manages core simulation processes, this includes processes for behavior, dynamics, image generation, driver I/O, experimenter I/O, and data collection.

2.14 METHOD USED

There are many of system either website-based or pc-based on the similar topic as my project, "Interactive 3D driving Test Simulation" but I choose this six for my literature review. The third, fifth and the last project, I have to admit that it is quite a competition for my future system on 3D driving test simulation. But I can summarize that these systems as list below:

- The cost of the system is too high for my project.
- Most of the system had a very specific target user such as teenager or high school student. This made the use of system very limited.
- The system developed is too advanced for the level of users in our country.
- This system provides an instructor which is able to monitor each student's progress from his own computer using the Performance Assessment System (PAS).
- The virtual driving provides a database differ to my project which is not including databases.

And I have to add that from the entire project, I obtain, there is one common purpose between my project and their project that is to provides an interactive diving test simulation to train users in 3D form.

What I actually found is, how that you actually capture the interest of the users is what important to the system and its purpose of existing. The systems have to look interesting or intriguing, it is like once that user used the system, and they will feel challenged by the system.

The navigation sequence is confusing for people to understand. We need an easy-to-use and user friendly system to maintain the users of the systems.

2.15 PROPOSED SYSTEM

The system that I'm going to develop can be engage as an interactive 3D driving tool to familiarize all level of users with driving test environment. It is create an effectives and interactive program for users. This is because my target users are from different backgrounds of users such as novice user, and professional users (instructor and traffic officer).

This system is a pc-based system which is a courseware (educational software) that can be used at driving school to train users. Interactive 3D Driving Test Simulation is a risk free 3D driving test simulation for PC. It is totally safe for novice driver, and in the same time it is a great tool for professional to increase their skill in driving.

Interactive 3D driving Test Simulation can be regarded as a courseware because it delivers driver education using 3D technology. Interactive 3D Driving Test Simulation can be implemented by driving schools, or it can be purchase personally for self-training.

This system is definitely an upper hand that can be used at driving school. Driving school's administration can install this program in a few computers and assign instructors to conduct the class. Before this could happen, the instructor himself must master the program. Then he/she can conduct good training.

For every lesson, drivers have to practice using the given route. At the end of each training session, the system will generate a report on the driving test status, whether user have passed the test or not.

2.2 TECHNOLOGY REVIEW

2.21 DEVELOPMENT MODELS

Methodology

A methodology is a collection of procedures, techniques, tools and documentation aids which helps system developers in their task of implementing a new information system. It consists of a set of phases, which consist of a set of sub phases. This guides the developers to the choice of techniques at various stages in the project and helps them to plan, manage, control and evaluate info systems project.

There are many types of development model in the software engineering. Waterfall model, V-shape model, and prototyping model are three basic models which can be used in small project. Incremental model and spiral model are usually used in the large project. When the large project is divide into well-defined small project (or phase or stage or iteration), a small project can use waterfall, V-shape, or prototyping model. Which model is chosen for development depends on the organization.

Waterfall Model

Waterfall Model [21] is a model that implies one development stage should be completed before the next begins. The waterfall model present a very high-level view of what goes on during development and it suggest to developers the sequence of events they should expect to encounter.

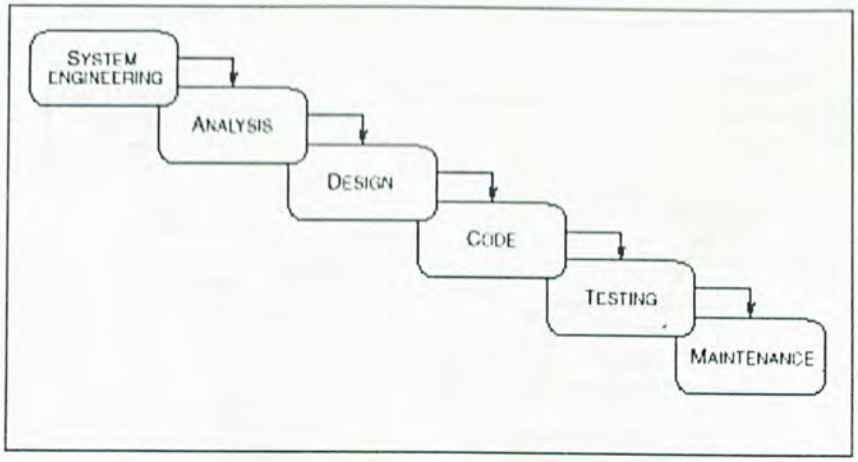


Figure 2.21: Waterfall Model [37]

The waterfall model can be very useful in helping developers lay out what they need to do. Its simplicity makes it easy to explain to customers who are not familiar with software development. Many other, more complex models are really just embellishments of the waterfall, incorporating feedback loops and extra activities. The biggest problem with the waterfall model is that it does not reflect the way code is really developed. Thus, the actual software development process, if uncontrolled, developers may thrash from one activity to the next and then back again, as they strive to gather knowledge about the problem and hoe the proposed solution addresses it.

V Model

V model [21] is a variation of the waterfall model that demonstrates how the testing activities are related to analysis and design (German Ministry of Defense, 1992).

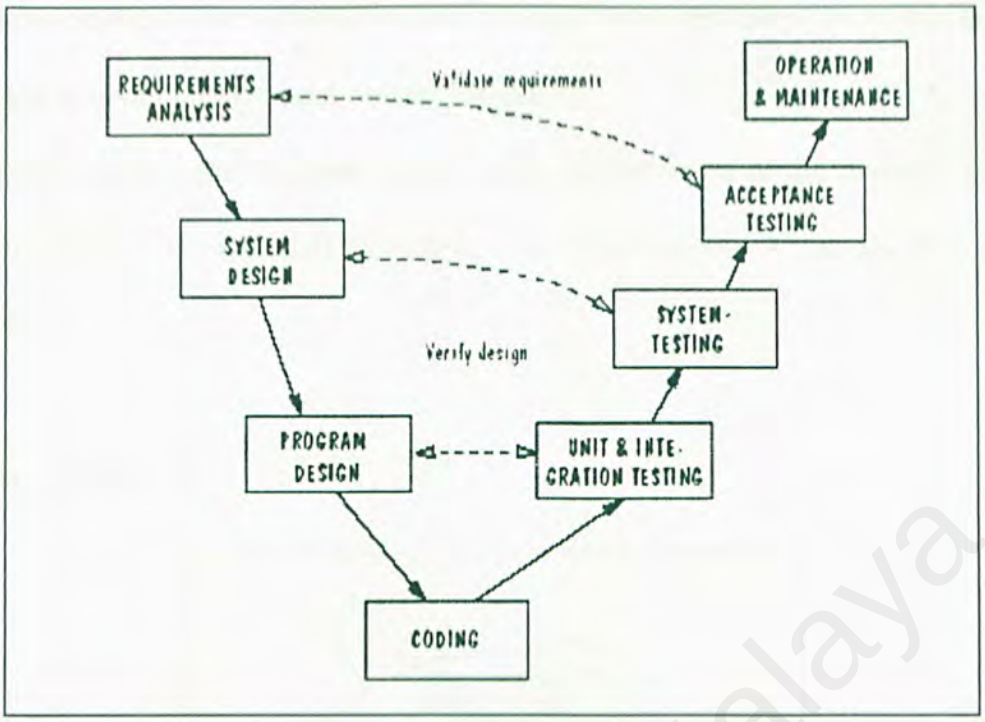


Figure 2.22: V Model [37]

As shown in figure 2.22, coding forms the point of the V, with analysis and design on the left, testing and maintenance on the right. Unit and integration testing addresses the correctness of program. Similarly system testing should verify the system design, making sure that all system design aspects are correctly implemented. Acceptance testing, which is conducted by the customer rather than the developer, validates the requirements by associating a testing step with each element of the specification. This type of testing checks to see that all requirements have been fully implemented before the system is accepted and paid for.

The model's linkage of the left side with the right side of the V implies that if problems are found during verification and validation, then the left side of the V can be re-executed to fix and improve the requirements, design, and code before the testing steps on the right side are reenacted. When we plan the requirement, we also plan for system testing.

Therefore, when the system is built, we have a whole set of test cases for system testing. By that way, the system does not meet user requirements.

In other words, the V model makes more explicit some of the iteration and rework that are hidden in the waterfall depiction. The focus of the V model is activity and correctness.

Prototyping Model

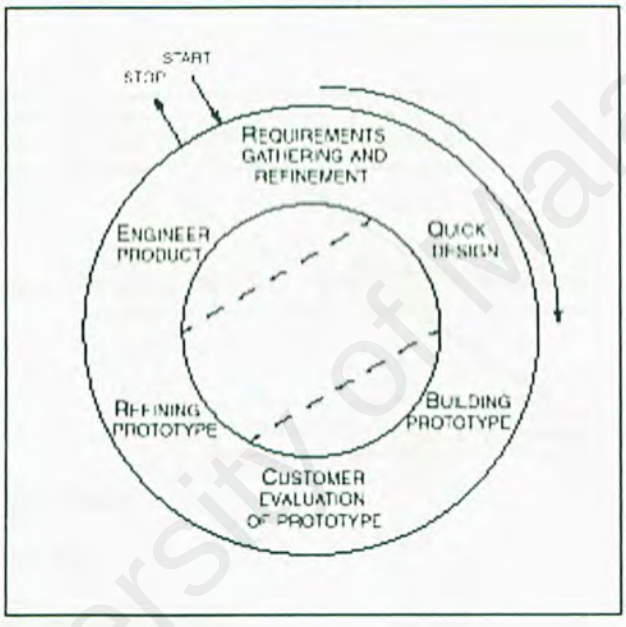


Figure 2.23: Prototyping Model

Prototyping model [21] allows all or part of system to be constructed quickly to understand or clarify issues, it has the same objective as an engineering prototype, where requirements or design require repeated investigation to ensure that the developer, user, and customer have a common understanding both of what is needed and what is proposed. The overall goal of prototyping model remains the same, specifically reducing risk and uncertainly in development. The initial design is revised until the developers, users, and customers are happy with the result.

The prototype model is also good for deploying the new technology. Before the technology is used, users are interested in know whether the technology works or not. Therefore, the prototype is a neat way to demonstrate the idea to users or customers.

Incremental Model

The designers develop the software in a number of stages and are able to deliver the product early. At each phase the designers have a goal to deliver certain features to customers. Incremental model is good for fast delivering product to the marker place.

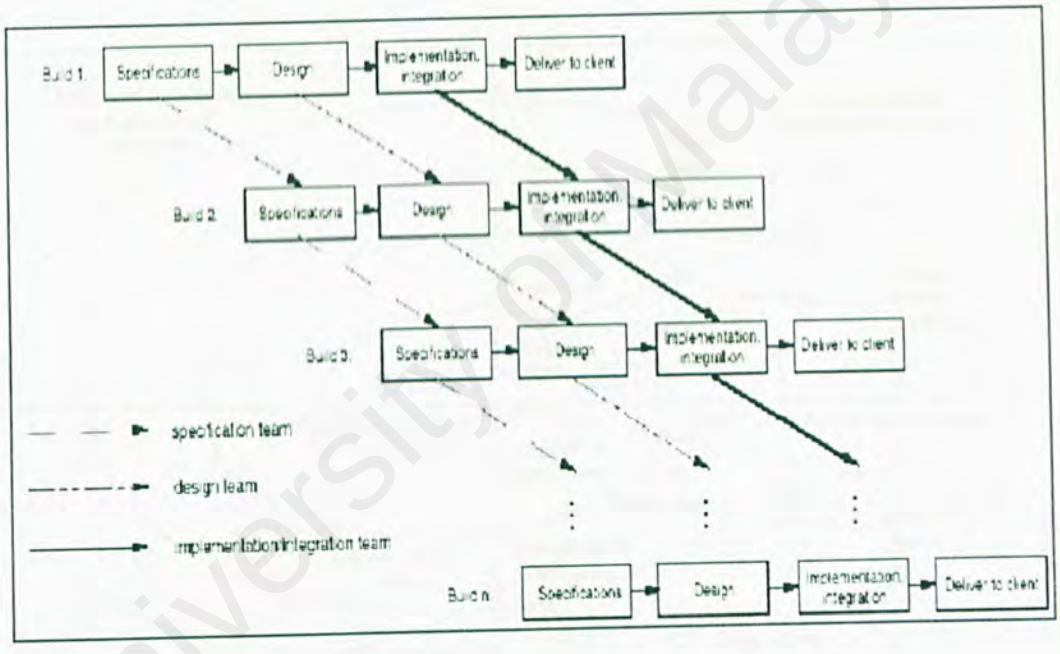


Figure 2.24: Incremental Model

Incremental model [21] has many advantages over the other techniques. One of advantages is that the system can be developed at several stages. Each stage has its own requirement; usually it has certain features or core of the system. Each stage can use V-shape, prototype or waterfall model to develop the requirement for this stage. Regardless

what kind of model is used in each stage, the product with certain features must be done at the end of the stage. Incremental model satisfies the requirement of fast delivery to the market place, so business people are interested in this model.

Spiral model

Boehm (1988) viewed the software development in light of the risk involved, suggesting that a spiral model could combine development activities with risk management to minimize and control risk.

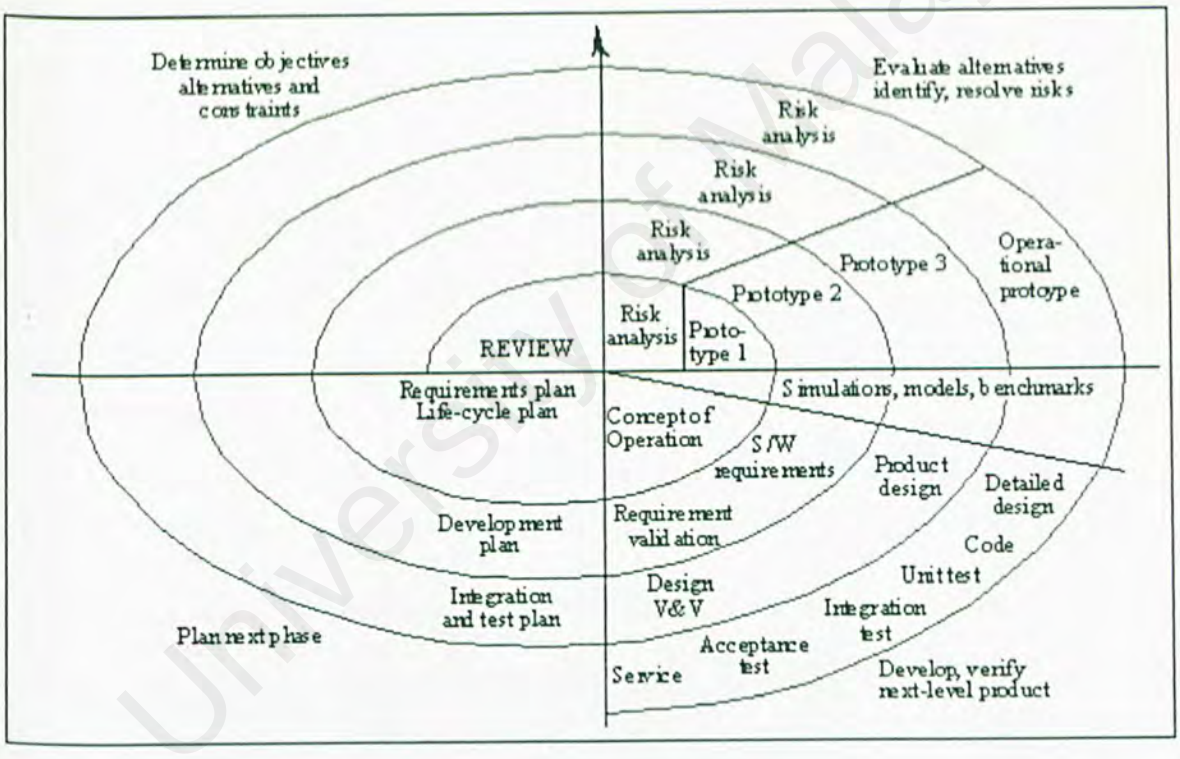


Figure 2.25: Spiral Model **[37]**

Spiral model [22] beginning with the requirements and an initial plan for development (including a budget, constraints, and alternatives for staffing, design, and development environment), the process inserts a step to evaluate risks and prototype alternatives before a "concept of operations" document is produce to describe at a high level

how the system should work. From that document a set of requirements is specified and scrutinized to ensure that the requirements are as complete and consistence as possible. Thus, the concept of operations is the product of the first iteration, and the requirements are the principle product of the second. In the third iteration, system development produces the design, and the fourth enables testing.

The risk analysis weighs different alternatives in light of the requirements and constraints, and prototyping verifies feasibility or desirability before a particular alternative is chosen with each iteration. When risks are identified, the project managers must decide how to eliminate or minimize the risk.

2.22 APPLICATION PLATFORMS

Table 2.21: Operating System statistic (http://www.w3schools.com/browsers/browsers_stats.asp)

Operating System	Jul 02	Oct 02	Jan 03	Apr 03	Jul 03
Windows 98 / ME	64%	62%	60%	59%	40%
Windows XP					19%
Windows 2000	20%	24%	27%	29%	30%
Windows 95	4%	3%	2%	2%	2%
Windows NT	5%	4%	3%	3%	2%
MAC	1%	2%	2%	2%	2%

∴Operating systems that count for less than 0.5% are not listed. [9]

Unix

Unix is an operating system developed at AT&T Bell Laboratories. Traditionally used on mini computers and workstations in the academic community has started to choose Unix for its openness. Unix, like other operating systems, is a layer between the hardware and the applications that run on the computer. Unix includes the traditional operating system components. It has functions that manage the hardware and the executing of applications separately. In addition, it includes a set of libraries, file system and process control.

Pros:

- It is a powerful and mature OS and network-based platform.
- Unix is not known only for its longevity and versatility as an operating system, but also for the variety and number of utility programs that called tool.
- Unix is the consistent way in which it treats files. It is very easy for the users to work with files because users do not need to learn special commands for every new task.

Cons:

- It is too expensive to use
- It need very powerful workstations and therefore not cost effective.

Linux

Linux is another version of Unix based OS. Linux is a free, from scratch operating system based heavily on the POSIX and UNIX API's. It supports both 32 and 64 bit hardware and provides a stable multi-user internet ready operating system. Linux itself is not UNIX; UNIX trademark is specific to systems that meet a complex set of X/Open standards and has a cost. Some Linux vendors however are working on "UNIX" branding.

Some of the many applications for Linux are: X11 Desktop, File Server, Computing Backend, Web Server, Usenet News, Terminal Server, FTP Archive, and Firewall.

Linux uses Internet and industry standard components and protocols giving a system with complete network integration. The operating system can act as a server for most major file serving protocols, and provide all the major Internet applications. The **X window system** provides a networked and platform independent graphical interface that (unlike proprietary user interfaces) allows one desktop to access applications running on multiple machines across local and wide area networks.

Pros:

- It is as stable as Unix
- It is developed under the GNU General Public License and its source code is freely available to everyone.
- Highly cost-effective ability to scale the size of the site as traffic grows.

Cons:

- It is developed by people worldwide, therefore lack proper organized support. It is more difficult to fund staff talented in any particular arbitrary combination of Linux / Apache / Jrun / Mod_Perl / PHP / Locomotive / whatever than it is to find staff talented in NT / IIS / COM.
- It is missing any pieces required to build a real application. Those pieces are problematic.
- Linux is inherently unsafe because every malicious cracker in the universe has the source code to the web site that develops under Linux.

Windows NT® Workstation 4.0

Windows NT Workstation 4.0 has the same interface as Windows 95. This means that you can use the same interface for all your Windows-based 32-bit desktops and servers. Windows NT Explorer displays your computer's contents as a hierarchy, or "tree", enabling you to see the contents of each drive and folder on your computer, as well as any network drives your computer is connected to. Windows NT Explorer replaces the File Manager, which was used in previous Microsoft Windows operating systems. The Task Manager is an integrated tool for managing applications and tasks, and providing key performance metrics of the Windows NT based system. Task Manager maintains detailed information on each application and process that is running on the desktop. It also provides a way to terminate applications and processes that are not responding, making the overall system more reliable. Numerous Accessibility Options are installed by default, making the system easier to use for people with disabilities.

Pros:

- The system would need to run on different hardware platforms with minimal changes.
- It could be locked down through software, meeting NSA's C2-level criteria.
- It would be POSIX-compliant, run existing Windows applications, and support open international standards.
- It would support symmetric multiprocessing (SMP).
- It could be easily expanded on by writing to a well-defined application programming interface (API).
- It could easily be ported to run in numerous different languages and writing systems, with minimal modifications to the software.

Cons:

- Require license fee.

Windows 2000 (W2K)

Windows 2000 is a commercial version of Microsoft's evolving Windows operating system. Previously called Windows NT 5.0, Microsoft emphasizes that Windows 2000 is evolutionary and "built on NT technology". Windows 2000 is designed to appeal to small business and professional users as well as to more technical and larger business market for which the NT was designed.

Windows 2000 provides an open, flexible environment for implementing powerful, highly customizable applications that share interfaces and other common elements that make them work together. It is an ideal platform for building custom applications because it contains strong web services with Internet Information Server (IIS) 5.0.

Windows 2000 professional is easier to deploy, manage, and support. Centralized management utilities, troubleshooting tools, and support for self-healing applications all make it simpler for administrators and users to deploy and manage desktop and laptop computers.

The rich features of Windows 2000 make it an ideal network operating system but it also does much more. It is the deal platform for enterprise computing, providing a unified, high performance environment for the web, for building applications, for streaming media for communications and collaboration.

Pros:

- Point-to-point Tunneling Protocol (PPTP) of Windows 2000 provides a way to use public data networks, such as the Internet, to create a virtual private network-connecting client PCs with servers. PPTP offers protocol encapsulation to support multiple protocols via TCP / IP connections and data encryption.
- System Policy Editor and User Profiles of Windows 2000 allow system administrators to manage and maintain users' desktops in a consistency manor. System policies are used for the standardization of desktop configurations and control the user work environment and actions.
- New application programming interfaces for server application developers and better server performance deliver improved throughput and scalability for server applications such as Microsoft SQL Server.

- The Task Manager of Windows 2000 is an integrated tool for monitoring applications and reports key performance metrics of the Windows 2000 system. It provides information on each application and process that are running on the workstation, as well as memory and CPU usage.
- Improved Windows 2000 Diagnostics Tool allowed for easy examination of the system resources such as IRQ, DMA and IO addresses, all presented in an easy-to-view graphical tool.
- Network Monitor of Windows 2000 is a powerful network diagnostic tool allows examining network and from the server at the packet-level. It allows capturing network of traffic for later analysis, this making troubleshooting network problems easier.
- Windows 2000 allows Object Linking and Embedding (OLE). It can combine the information from several applications into one compound document using the special OLE capabilities of Window-based application.

Cons:

- In order to set up an e-mail server, user will have to buy a separate software package.
- Insignificant changes to a Windows 2000 configuration require a shutdown and reboot in order to make the changes take effect. Change the IP address of the default gateway also has to reboot.
- The ongoing maintenance and support requirement of Windows 2000 can make them much more costly to run.

Window XP

Windows XP is built on an enhanced Windows 2000 code base, with different versions aimed at home users and business users: Windows XP Home Edition and Windows XP Professional. Windows XP integrates the strengths of Windows 2000 – standards-based security, manageability and reliability – with the best features of Windows 98 and Windows ME – Plug and Play, easy-to-use user interface, and innovative support services to create the best Windows yet.

Pros:

- ***Reliability*** - XP appears to be vastly more dependable than its predecessors, though the true test comes over the long haul. Its underpinnings are based on the solid Windows NT/2000 engine (used by businesses), and not outmoded MS-DOS architecture. When a program freezes or crashes, the whole system isn't brought to its knees. Reboots are rare.
- ***Straightforwardness*** - Microsoft serves up apologetic XP-compatible 2002 version of Quicken and instead received a "16 bit MS-DOS Subsystem" notice that the "NTVDM CPU has encountered an illegal instruction." Quicken otherwise behaved normally.
- ***Attractive*** - The desktop is easier on the eye. Task-bar buttons are organized into groups; the rarely used notification icons that used to clutter your system tray are hidden. The Start button is now green with a rounded edge. Click on it and a window appears. On the left are the Internet Explorer and e-mail icons, plus icons dedicated to the programs most often use.

- ***User accounts*** - User can set up a separate password-protected account for each member of your household with different access levels: "computer administrator" (able to make system wide changes, install or remove files and hardware) or "limited" (can alter only a few settings). Each person gets a unique My Documents folder and can arrange the screen layout any way that he or she wants.
- ***Network friendliness*** - XP includes simple wizards for setting up a home network, and it automatically recognized my speedy cable modem Internet service.
- ***A control panel under control*** - A simplified category view helps non-techies get at tasks easily. Under the "appearance and themes" category are listings for screen saver or desktop background changing.
- ***Smart folders*** In My Pictures, users can view thumbnail snapshots of their photo files. On the left side of the file pane are tasks such as creating slide shows or ordering prints online, from Microsoft partners such as Fujifilm and Shutterfly. In My Music, a mouse click lets you shop for tunes on the Net (not surprisingly, through Microsoft's Windows Media.com).
- ***Help from afar*** - A new feature called Remote Assistance lets user invite a techie pal to connect to user's PC to help solve problems.

Cons:

- ***Activation*** - User have 30 days to activate XP via the Net or over the phone before the OS stops working. Activation doesn't take long and is Microsoft's way of thwarting piracy. But it means consumers can't share XP with more than one machine in the house. And since Microsoft uses a complicated calculation to tag XP to a

specific machine, user will possibly have to reactivate it should they undertake a serious upgrade of PC's internal components.

- ***Functionality*** - Some software and hardware would not operate under XP, at least without updated patches or drivers.

2.23 PROGRAMMING LANGUAGE

2.231 GRAPHIC LIBRARY

C++

C++ is an object-oriented programming (OOP) language [23] that is viewed by many as the best language for creating large-scale applications. C++ is a superset of the C language.

A related programming language, java, is based on C++ but optimized for the distribution of program objects in a network such as the Internet. Java is somewhat simpler and easier to learn than C++ and has characteristics that give it other advantages over C++. However, both languages require a considerable amount of study.

The C++ programming languages is an extension of C that was developed by Bjarne Stroustrup in the early 1980s at Bell Laboratories. C++ provides a number of features that "spruce up" the C language, but more importantly, it provides capabilities for object-oriented programming.

Significant Language Features

Object-oriented programs are easier to understand, correct and modify. Many other object-oriented languages have been developed, including most notably, Smalltalk. The best features of C++ are:

- C++ is a hybrid language-it is possible to program in a C-like style, an object-oriented style, or both.
- C++ programs consist of pieces called classes and functions. You can program each piece you may need to form a C++ program. The advantage of creating your own functions and classes is that you will know exactly how they work. You will be able to examine the C++ code.

Areas of Application

C++ provides a collection of predefined classes, along with the capability of user-defined classes. The classes of C++ are data types, which can be instantiated any number of times. Class definitions specify data objects (called data members) and functions (called member function). Classes can name one or more parent classes, providing inheritance and multiple inheritances, respectively. Classes inherit the data members and member functions of the parent class that are specified to be inheritable. Therefore it is mainly used for:

- Software Engineering
- Graphics

Aadvantages:

C++ is a general purpose programming language with a bias towards systems programming that are:

- a better C
- supports data abstraction
- supports object-oriented programming
- Supports generic programming.

Drawback:

- It is much more difficult to start, you not only have to learn the language, but you have to choose and learn class libraries.
- Users are much more likely to pick up bad programming habits, if you are using C++ with its legacy features and mishmash of class libraries. You have to do all the memory resource management yourself.
- You will probably spend more time debugging the faults caused by the factors above, when your time should be spent on learning the concepts.

OpenGL

OpenGL is a cross-platform standard for 3D rendering and 3D hardware acceleration. The software runtime library ships with all Windows, MacOS, Linux and Unix systems.

- **OpenGL delivers fast and complete 3D hardware acceleration.**

  Today's applications and games manipulate massive amounts of data in real-time by using OpenGL hardware accelerated geometry, real-time lighting, clipping, transformations and rendering

- **OpenGL makes real-time 3D effects possible**

  OpenGL hardware acceleration adds detail & special effects to images without compromising performance. Examples include real-time fog, anti-aliasing, volume shadows, bump mapping, motion blur, transparency, reflections, 3D textures, volume rendering and more

- **OpenGL is designed to support future innovations in software and hardware**

  The extension mechanism in OpenGL can expose radically new hardware/software features that simply never existed when OpenGL was originally conceived. So you can be confident that you will continue to get the best possible performance from your applications and games as hardware technology advances

- **OpenGL is on every platform**

  Ever wished that great Unix application would move to Windows or a great Windows game to Linux. Since OpenGL is fully cross-platform, it is relatively easy to port games or applications from one platform to be moved to another. This greatly expands the wealth of applications and games that support OpenGL.

- **OpenGL is stable.**

  High End 3D workstation & supercomputers users have been enjoying OpenGL since 1992. Today, OpenGL is also the most popular and powerful 3D solution for Windows, Mac and Linux-based computers, for professional and consumer use.

Areas of application:

- **Games**

  OpenGL is the essential 3D library that lets you harness the power of hardware acceleration to thrust games into the realm of professional-quality rendering, texture-mapping and special effects. Almost all of the leading games (e.g. Quake III, Half-Life, MDK2, Baldurs Gate, Decent 3, Madden NFL 2001 etc.) require OpenGL for hardware acceleration.

- **Professional 3D graphics & effects**

  OpenGL is the 3D power behind all of the incredible 3D graphics & effects you see on TV and at the movies. (Think Star Wars). Hardware that uses OpenGL makes the compelling 2D and 3D graphics in broadcasting, CAD/CAM/CAE, entertainment, medical imaging and virtual reality. All leading 3D modeling, rendering & animation and visualization software use OpenGL.

Platform

- **Windows**

  An OpenGL v1.1 software runtime ships as part of Windows NT v4.0, Windows 2000, Windows 98, and Windows 95 (OSR2). The OpenGL v1.1 libraries are also available as the self-extracting archive file from the Microsoft Site via HTTP or FTP.

  OpenGL v1.2, 1.3, & 1.4 are included with the drivers for your OpenGL video cards. So users only need to make sure they have the latest OpenGL driver for their video card. If users do not have the latest driver with OpenGL 1.2 or 1.3 or 1.4 supports, go

to the website of your video card manufacturer download the latest OpenGL driver for your card and OS.

- **Mac OS**

  OpenGL ships with OS 9 and OS X. It obtains the latest software version on the Apple OpenGL web site.

  Also need an OpenGL hardware accelerator driver. All new PowerMac G4, iMac, iBook and PowerBook computers ship with built in hardware acceleration and include the correct hardware driver. If users buy a different or additional hardware board, they can obtain the driver from each board manufacturer's web site.

- **Linux**

  OpenGL and GLX and the X server integration of GLX, are Linux system components, and should be part of the Debian, RedHat, SuSE, or Caldera distribution. Users also need an OpenGL hardware accelerator driver. Drivers for OpenGL 3D hardware acceleration are available from NVIDIA, 3dfx, Matrox and ATI. XiGraphics provides commercial OpenGL drivers for professional applications. XFree86 4 includes DRI GLX drivers for hardware-accelerated OpenGL.

DirectX

Microsoft DirectX is a group of technologies designed to make Windows-based computers an ideal platform for running and displaying applications rich in multimedia elements such as full-color graphics, video, 3-D animation, and surround sound.

Microsoft DirectX is an advanced suite of multimedia application programming interfaces (APIs) built into Microsoft Windows operating systems. DirectX provides a standard development platform for Windows-based PCs by enabling software developers to

access specialized hardware features without having to write hardware-specific code. This technology was first introduced in 1995 and is a recognized standard for multimedia application development on the Windows platform.

At the core of DirectX are its application programming interfaces, or APIs. The APIs act as a kind of bridge for the hardware and the software to "talk" to each other. The DirectX APIs gives multimedia applications access to the advanced features of high-performance hardware such as three-dimensional (3-D) graphics acceleration chips and sound cards. They control low-level functions, including two-dimensional (2-D) graphics acceleration which is support for input devices such as joysticks, keyboards, and mice and control of sound mixing and sound output. Because of DirectX, what you experience with your computer is better 3-D graphics and immersive music and audio effects.Windows 98, Windows 98 SE, Windows Millennium Edition (Windows Me), Windows 2000, Windows Server 2003, and Windows XP support DirectX 9.0.

Python

Python is a powerful object-oriented programming language initiated by Guido van Rossum in 1990. It is easy to start writing simple programs in Python, and even using classes and objects involves a relatively gentle learning curve. Python is cross-platform (Windows, Mac, Linux). It is open source (and therefore free) and supported by a sizable developer community. Python is a general-purpose programming language with significant numerical capabilities, though a common use of Python is managing servers, administering systems, and others.

Real-time 3D Graphics: Visual

Python itself does not provide graphics output. The Tk graphics library can be used to create 2D graphics, but it is difficult to use. In the spring of 2000, while a sophomore in computer science at Carnegie Mellon University, David Scherer created a 3D graphics module for Python, called "Visual," that is exceptionally easy to use. A program can create 3D objects (such as spheres, curves, etc.) and position them in 3D space. Visual, running in a separate thread, automatically updates a 3D scene many times per second, to reflect the current positions of the objects. The programmer does not need to deal with display management, but can focus on the computational aspects of the program. The user can navigate in the 3D scene by using the mouse to zoom and rotate while the program is running. Visual supports full vector algebra. VPython focus on the physics computations without having to write explicit graphics statements yet obtain 3D visualization. Users can do true vector computations, which improves their understanding of the utility of vectors and vector notation.

Platforms Supported

The Python language itself is cross-platform. The Visual module, and the idle integrated development environment, works with Python:

- Windows
- Unix
- Linux
- MacOSX.

C programming

The C programming language [24] was devised in the early 1970s as a system implementation language for the nascent Unix operating system. Derived from the type less language BCPL, it evolved a type structure; created on a tiny machine as a tool to improve a meager programming environment, it has become one of the dominant languages of today.

Advantages:

- Get more speed than C++ (not having built in Object Oriented Support).

Disadvantages:

- Not having built in Object Oriented Support.

Pros & Cons	OpenGL	DirectX
Pros	▪ Portability Allows programmer to write application that will run on multiple platforms. ▪ Open standard Companies with a platform can buy a license and create their own implementation wit it. ▪ Many features and Extensions. Many companies have created extension for use of such areas as military, CAD, and of course games ▪ Extensive Documentation and resources. Many sites included www.opengl.com are available that include tutorials and class definitions.	▪ Programming Pixels and Vertex Shaders Allows users to create custom code for portions of rendering pipeline ▪ Object-oriented programming friendly Uses COM (Computer Object Model) programming ▪ Being platform has advantages Do not have to write vendor specific code.

Cons	▪ Too many extensions The code can become cluttered and confusing. ▪ Most video cards have support for OpenGL and DirectX, but a few still have les support for OpenGL. ▪ Not object oriented. Can be a problem for a growing popularity of OOP	▪ For use in win32 application Not too great for portability ▪ Only updated about once a year. Hard to say on the cutting edge ▪ Must be familiar with COM programming difficult for graphic beginners not familiar with COM ▪ Lengthy and sometimes confusing code COM programming involves a lot of pointers.

Table 2.22: Comparison between OpenGL and DirectX

2.24 DEVELOPMENT TOOLS

2.241 APPLICATION PROGRAMMING INTERFACE

3D Studio Max

The full-featured 3D Studio Max 2.5 [34] animation package is the graphics equivalent of a major office suite. Few office users need all the options in a suite, but graphics professionals who work on different projects need all the options they can get.

Graphics pros likely will grasp the basics of Autodesk Inc.'s Studio Max without studying the 5-inch-thick pile of documentation. The stack would be even higher if the newest features were not covered only by online documentation.

Except for the interface, 3D Studio Max for Microsoft Windows 95 and Windows NT has almost nothing in common with MS-DOS versions. Everything happens in the main four-window editing screen.

3D Studio Max could do almost every graphics job well, from basic object design to photorealistic animation.

Advantages:

- NURBS support, key frame animation and multiprocessor support
- Excellent mapping and editing
- Auto backup and auto filename increment

Disadvantages:

- Too many hardware locks
- Ray-trace rendering requires Radi Ray package

MAYA

Academy Award winning Maya 3D animation and effects software is at the forefront of technological innovation. Its development has been inspired by the film and video artists, computer game developers and design professionals who use it daily to create engaging digital imagery, animation and visual effects.

The Maya Family **[35]**

Maya Unlimited™ - The world's most powerful 3D animation and visual effects software and the ultimate version of Maya. Maya Unlimited is available on Windows XP Professional, Windows 2000 Professional, SGI IRIX and Linux

Maya Complete™ - Maya Complete makes the foremost 3D content creation tools accessible to a broad range of computer graphics professionals in the film, broadcast, industrial design, visualization, game development and web design industries. It is the leading full 3D production solution. Maya Complete is available for Windows 2000 Professional, Windows XP Professional, Mac OS X, SGI IRIX and Linux operating systems.

Maya Personal Learning Edition ™ - Maya Personal Learning Edition™ is a special version of Maya that provides free access to Maya Complete™ software for non-commercial use. Available for Maya 5 on the Windows 2000 /XP Professional and Mac OS X operating systems.

Maya Personal Learning Edition 5 - Maya Personal Learning Edition 5 contains almost every feature found in the full commercial version of Maya Complete including these new additions:

- A new hardware rendering option takes advantage of the power of next-generation graphics cards to produce near software-quality images at often dramatically faster speed.

- A unified rendering workflow provides easy and consistent access to the Maya rendered through a common interface.

- Animation enhancements to constraints, forward/inverse kinematics and ghosting give animators new levels of flexibility.

- Maya Paint Effects(TM) to polygon conversion offers a whole new range of looks, as well as more editing and output options.

- An improved polygon reduction method is valuable to users making lightweight game models and MODs.

Maya Personal Learning Edition is for everyone, from creative professionals and hobbyists who have never tried 3D to advanced professionals seeking to enhance their skill set. In addition to the Learning Maya(TM) Beginner's Guide, Alias(R) has made learning easier than ever with enhanced resources. Improvements have been made to numerous online help features including: an improved Maya 5 online help search engine, that supports both phrase and Boolean searching; and a new guide titled, Extra Learning Resources, that presents a series of recommended steps to learning Maya and suggests self-paced Learning Tools and resources.

In a continuing program to regionalize content, Japanese documentation is available for Maya Personal Learning Edition on both the Microsoft(R) Windows(R) and Mac(R) OS X platforms. The experience of learning Maya is multi-lingual as Alias presents the Learning Maya Beginner's Guide, containing a one hour instructional DVD, an introduction to Maya booklet and a Maya Personal Learning Edition 5 CD with English, French, German, Japanese and Korean language tracks. Also new in this version of Maya Personal Learning Edition, is a one-step documentation installation in either English or Japanese.

The Maya Personal Learning Edition restricts users to non-commercial applications through the display of a watermark on images as well as through the use of a special non-commercial file format. Maya Personal Learning Edition does not ship with the Maya API and cannot be used with standard plug-ins. Use of the software is restricted to personal, non-commercial applications only: use in schools and training center classrooms/labs or courses is considered commercial use. As such, educational institutions and training facilities should contact their local sales office for more information about Alias' extensive academic and training center programs.

Maya System Requirements

Software Requirement

Maya requires one of the following operating systems:

- Microsoft XP Professional, Microsoft Windows 2000 Professional, (Service pack 2 or higher)
- SGI IRIX 6.2.15
- RedHat™ Linux 7.3 or 8.0
- Apple Mac OS X 10.2.4 or higher (Maya Complete ONLY)

Maya requires one of the following browsers:

- Microsoft Internet Explorer 4.0 or higher
- Netscape 7.0 or higher

Hardware Requirements

A minimum, Maya requires a system with:

- Intel® Pentium® II or higher, AMD Athlon™ processor
- 512 MB RAM
- CD-ROM Drive
- Hardware-Accelerated OpenGL graphics card
- 3-button mouse with mouse driver software
- 450 MB of hard disk space

VRML

VRML (**V**irtual **R**eality **M**odeling **L**anguage) [18] is a hierarchical scene description language that defines the geometry and behaviors of a 3D scene or "world" and the way in which it is navigated by the user. VRML world files have the file extension .wrl (or .wrz for gzip compressed files) and require either a stand-alone application or web browser plug-in to be viewed.

VRML is the only standardized (ISO/IEC 14772) 3D format suitable for Web delivery (due to its generally small file size). It is versatile, compact, extensible and constantly evolving - theoretically there is no limit to what can be achieved with VRML given sufficient computing power.

Features

- **Navigation** - many control options - walk, fly, examine or none
- **Viewpoints** - pre-defined camera positions
- **Models** - primitives (box, sphere, cone, cylinder), extrusions, indexed face set (mesh), line set, point set, elevation grid and text
- **Materials** - diffuse colour, specular, emissive, ambient, shininess, transparency, colour per vertex
- **Sound** - fully specialized 3D audio in WAVE or MIDI format
- **Textures** - support for JPEG, GIF, PNG and MPEG1 video. (support for Flash, RealMedia, AVI, multi-texturing and environment mapping)
- **Lighting** - directional, point or spotlight
- **Special Nodes** - background, switch, hyperlink (anchor), billboard and fog
- **Performance** - LODs (levels of detail), visibility distance culling
- **Collisions** - collision detection between user and objects
- **Animation** - animate position, rotation, scale, points, colors and much more. Scope for many separate animations in one world all with different time lines and triggered by different events. Almost every attribute can be animated!
- **Sensors** - sense user activity such as touch, drag (plane, cylinder and sphere), time, proximity, visibility
- **Routes** - scripts, animations and object properties can be "wired" together in an infinite number of ways to create any effect
- **Compact** - extremely small file size with gzip compression
- **Modular** - references external textures, models, scenes and scripts

- **Extensible** - if the core VRML nodes aren't enough you can create your own nodes using Prototypes
- **Other** - Support full-screen mode and 3D stereo with the right hardware. They can also both be embedded in either a web page or a stand-alone application with full control of the scene graph and 3D engine from C++, java, javascript and vbscript.

New functionality and formalizes some extensions that have developed since VRML97

- Full backwards compatibility with VRML97
- Full extensibility through Components and Profiles
- Core profile is a limited subset of VRML97 for lightweight plug-ins and MPEG4 compatibility
- Optional XML and binary encodings, in addition to VRML's traditional ASCII encoding
- **H-Anim** - Humanoid Animation. Standardized hierarchical structure and interface for animated avatars
- **GeoVRML** the representation of accurate real-time geographical data using VRML
- **NURBS** - organic models described using NURBS mathematics rather than polygons thus significantly reducing file size and increasing render optimization
- **2-way Web page communication** through a multi-language scene authoring interface (SAI)

- **Universal Media** - In-built standard media items (textures, models etc) to speed download time
- Multi-texturing extensions
- Enhanced sensors (keyboard, joystick, drag 'n' drop etc)

LightWave

LightWave, provides art director, cinematographer and director, all rolled into one that can make user feel like a director. Their actors consist of objects that they build in Modeler or import from another program. User place lights to illuminate them, assign surface colors and values to *clothe* them, and then choreograph their movement to animate them. Once user has their actors in place, a camera records all of their movements, creating images of their movements that they can compile into animations.

LightWave can create animations and still images for virtually any use, from tiny 16 x 16 pixels Web graphics, all the way up to a resolution of 16,000 x 16,000 pixels, and anywhere in between. A multitude of preset resolutions are also provided, including industry standard D1 and D2 (NTSC and PAL) video resolutions.

User create animations by generating a single frame at a time, then recording each frame to some type of playback device, like videotape or a hard-disk playback device, or even film. Frames can also be saved in computer animation format, like AVI.

SOFTIMAGE|3D

SOFTIMAGE|3D is more to game creation. The SOFTIMAGE|3D games development environment provides a selection of platform-specific tools and exporters for on-target platforms like Sony PlayStation, PC/DirectX and Nintendo 64. Take maximum advantage of platform and rendering options with unbeatable on-target viewing tools and target-specific rendering attribute editor.

SOFTIMAGE|3D also features import and export of the Softimage dotXSI v.3.0 file format, designed especially for interactive media applications. The dotXSI file format allows ASCII import and export of characters, models, and animation for complete customizability of any game development pipeline. SOFTIMAGE|3D offers vertex color manipulation and authoring, including alpha-channel support, as well as powerful UV texture editing, and texture pre-lighting with Render map to capture sophisticated mental ray lighting and effects directly in texture maps.

- ***Build character***

  SOFTIMAGE|3D modeling is all about artistic workflow. The polygon editing mode provides direct access to the most commonly used polygon creation tools, such as ray-cast selection and extrusion. A complete suite of polygon reduction techniques is at your disposal, keeping you safely within the constraints of your polygon budget. And when the basic modeling is done, you'll find a range of quick and effective tools for advanced manipulation, creating constraints, and deforming surfaces.

- ***Create games***

  Realistic games require top-notch texturing. SOFTIMAGE|3D's powerful UV editing controls let you directly manipulate UV textures – flip, rotate, SRT, or even move the

center to get the precise look you want. Work with the integrated 3D paint system to blend textures together, and save the result to an integrated clipboard designed to help manage textures and palettes. SOFTIMAGE|3D supports vertex color manipulation. Go ahead and paint directly onto vertices. A full suite of color reduction tools provides palette generation control, including alpha-channel support. Texture pre-lighting allows you to minimize or even remove the lights in a scene – blend lights into textures, or generate a completely separate illumination map to be processed in real-time by your game platform.

- ***Create Life***

  Characters are more than a mixture of polygons and textures. You need to bring them to life, and that requires an advanced set of animation tools. Softimage animation environment gives user more time to tune and perfect character movements. Use constraints, expressions and function curve tools to position and drive the animation exactly as user wish. Apply fast, intuitive inverse kinematics (IK) chains and envelopes models to produce a complex and realistic character.

Criteria	**Maya Complete 4.5**	**3D Studio Max**	**Lightwave 3D**
Producer	Alias/Wavefront	Autodesk	New Tek
▪ Category	3D rendering and animation	Modeling, Rendering, Animation	Rendering, Animation.
▪ System	Windows XP, Windows 2000 professional, IRIX, Linux, and Mac OSX	Win95/98, NT	Winn95, NT, IRIX, Mac, Solaris, 32MB RAM, 10 MB HD space.

▪ Features 1.View /Base of use	Creates three orthographic views-top, side, and front (In what call a " Four View windows) Easily navigate via the same keyboard and mouse combination for zooming, tracking and rotation. The fact that Maya is build on Maya embedded language (MEL) is one of the program's most powerful features.	The default view uses four different view ports; top, front, left and perspective. The views can be manipulated using the mouse and ctrl-, shift and alt keys. Has its own scripting language which allows users to modify their work.	The views are flexible and the setting up a snap grid is easy. Lightwave does not let the views be controlled using mouse modifier keys, but the smooth on screen control make the views easy to work.
2.Primitives/Polygon Creation Tools	Simple 3D object: Polygonal or nurbs primitive, camera orthographic, lights, and manipulator control.	The most common primitive can be created in 3DStudion Max: 2D-shapes and 3D-objects. Camera, lights, and particles are also easily created and manipulated.	Lighwave has a control window so that the position, orientation and every aspect of any object can be controlled. It is also possible to create.
3.Editing Polygon	Provides many different of editing polygon such as displaying polygons, displaying faces, coloring vertices, selecting polygon, moving, extracting and separating polygons.	Unwanted objects can be easily hidden which is really helpful. It is possible to select a single point, a face, an entire object etc. It is fairly intuitive to select, deselect and manipulate existing objects.	Selecting object is a bit difficult but there are plenty of options for moving and rotating objects.

3.Complex surface And volume operations.	Maya has numerous tools for creating and manipulating surfaces. Using subdivision surfaces for modeling, texturing, animation, and rendering. Maya interfaces also offers 10 tools in Maya artisan to provide one of the most intuitive ways to model, select and edit geometry.	3DStudio Max also has numerous tools for creating and manipulating surfaces. From the basic extrusion and twisting advanced tools like softness (less impact on control points far from the selection). Program is a mixture between a modeler, an object-manipulator anda file-converter, and it is available for several hardware platforms.	Lightwave offers lots of interesting options for manipulating 3D objects. Using these tools is particularly easy.
5.Texture Maps	Too many options. For example Maya visor's panel offers 26 textures: ten 2D textures, eleven 3D textures, five environment textures, and a new layered texture.	Good, but too many option, which makes it a bit complicated to apply the texture as intended. But it is probably just the matter of getting used to the program.	Lightwave's texture mapping interface is good, but like many others it is complicated and difficult to figure out what the results will actually be.

Table 2.23: Comparison between Maya, 3D StudioMax and Lightwave

2.3 CHAPTER SUMMARY

All these researches were done mainly to gain information for this project. The information gathered includes the concepts and strategies to develop an Interactive 3D Driving Test Simulation, reviews on the existing systems applications, development methodologies, and development tools.

Research on Interactive 3D Driving test Simulation's concepts and strategies is to have a better understanding on the requirements of this project. The review on the existing systems applications were done by browsing to some of the Application Service Provider web site. Assessing current existing systems or applications allows identification of the weaknesses that are to be overcome in this project. Meanwhile the strengths of existing systems were studied so that it can be adapted into this project.

As for the development methodologies, this review of literature focuses mainly on five development models that are the Waterfall model, V-shape model, Prototyping, Incremental model and Spiral model. Each modal has its own features, which are different from various resources.

The information gathered in development tools for the project was analyzed. All these information was obtained from the internet using search engines and other resources. Information on six different server platforms such as Microsoft XP, Microsoft Windows NT, Microsoft Windows 2000, Linux and Unix had been gathered and analyzed. Each platform has its strengths and weaknesses.

The information gathered on programming languages such as C programming, C++, OpenGL and Python are also being analyzed. As for the Development tools, information was

gained from sources like books and Internet. All the development technologies and programming languages chosen for this project will be mention in next chapter.

Literature review is the stepping stone for me to consider the requirements for my proposed project. By reviewing the math systems, I manage to find some important features from those systems and the features gave me some idea to enhance my proposed project.

After research and technology consideration, the proposed project – Interactive 3D Driving Test Simulation is feasible to develop.

CHAPTER 3

METHODOLOGY

Systems developments methodology is a standard process used by an organization to complete all of necessary steps to analyze, design, implement, and maintain the information system. Standardization of the process provides several benefits to an organization. Benefits include consistency of systems throughout the organization for integration of operations, control of process integrity, and greater efficiency in systems development. Standardization allows economies of scale regarding resources used in development and maintenance of systems. So, the tools and people can easily be reallocated to different projects (project teams can be formed quickly; transition to a new project is more efficient).

3.1 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

System Development Life Cycle (SDLC) [15] is the overall process of developing information systems through a multi-step process from investigation of initial requirements through analysis, design, implementation and maintenance. There are many different models and methodologies, but each generally consists of a series of defined steps or stages.

SDLC refers to a methodology for developing systems. It provides a consistent framework of tasks and deliverables needed to develop systems. The SDLC methodology may be condensed to include only those activities appropriate for a particular project, whether the system is automated or manual, whether it is a new system, or an enhancement to existing systems. The SDLC methodology tracks a project from an idea developed by the

user, through a feasibility study, systems analysis and design, programming, testing, implementation, and post-implementation analysis. Documentation developed during the project development is used in the future when the system is reassessed for its continuation, modification, or deletion.

Although the SDLC is a complete structure, the length of time required to complete the steps involved, as well as whether one step must be completed before another is started, depends on several factors. These factors include the level of the system, the time available to develop the new system, and the complexity of the tasks being affected. In any case, the five steps, or phases, of the systems development life cycle are these:

- Preliminary investigation
- Systems analysis
- Systems design
- Systems development/acquisition
- Systems implementation and maintenance

The general nature of these phases is the same regardless of the level of the system being developed. The specific activities in a phase, however, will vary.

3.2 METHODOLOGY CONSIDERATION

A methodology is a collection of procedures, techniques, tools and documentation aids which helps system developers in their task of implementing a new information system. It consists of a set of phases, which consist of a set of sub phases. This guides the developers to the choice of techniques at various stages in the project and helps them to plan, manage, control and evaluate info systems project.

The main objectives of following a methodology is to make the development cycle as efficient as possible, to complete development within lowest possible cost keeping the highest quality, and to achieve the fastest turn-around. Another important objective is to make future maintenance easier and faster. The development cycle for each and every project is some way unique, depending on system requirements and their unique operating environment. Design and development methodology also varies depending on the software, hardware technologies chosen.

3.21 ADVANTAGES OF GOOD METHODOLOGY

A good methodology that able to provide the effective ways of system development is best defined before the project starts and then becomes the framework to development staff.

Some advantages offers by a good methodology:

- Provides a standard framework that the developer does not have to reinvent the wheel for each project.
- Each method or tool in the methodology results in successful completion of each development task.

- Reviews procedures are available to identify any errors, inconsistencies and discrepancies during development.
- Increase the system quality by forcing the developer to produce flexible systems and adequate documentation.
- Provides better understanding of user needs and validation of user needs.
- Provides the management with tools to review project progress and checklist to access tasks and deliverables.
- Improves communication among management, analyst, programmers, users and other stakeholders by providing a communication base.
- Facilitates planning and controlling the project.
- Ensures that a consistent reproducible approach is applied to all projects
- Reduces the risks associated with shortcuts and mistakes
- Produces complete and consistent documentation from one project to the next
- As development teams and staff change over time, the result of prior work can be easily retrieved, understood and modified as required

3.22 CONCLUTION ON DEVELOPMENT METHODOLOGY

The methodology used in the development of Interactive 3D Driving Test Simulation is the Waterfall model. As the name imply the stages of development cascades from one phase to another. Each stage of a development is required to be completed before proceeding to the next phase. The Waterfall model offers the benefit of a structured development, in addition to good visibility and proper documentation for each development stage.

The Waterfall Model is the earliest method of structured system development. Although it has come under attack in recent years for being too rigid and unrealistic when it comes to quickly meeting customer's needs, the Waterfall Model is still widely used. It is attributed with providing the theoretical basis for other Process Models, because it most closely resembles a generic model for software development.

The following diagram gives a visual layout of the waterfall method which will represent the project system development processes. Each stage must be completed before moving on to the next. Testing is done in each stage and the developer may go back to the previous stage to correct any errors.

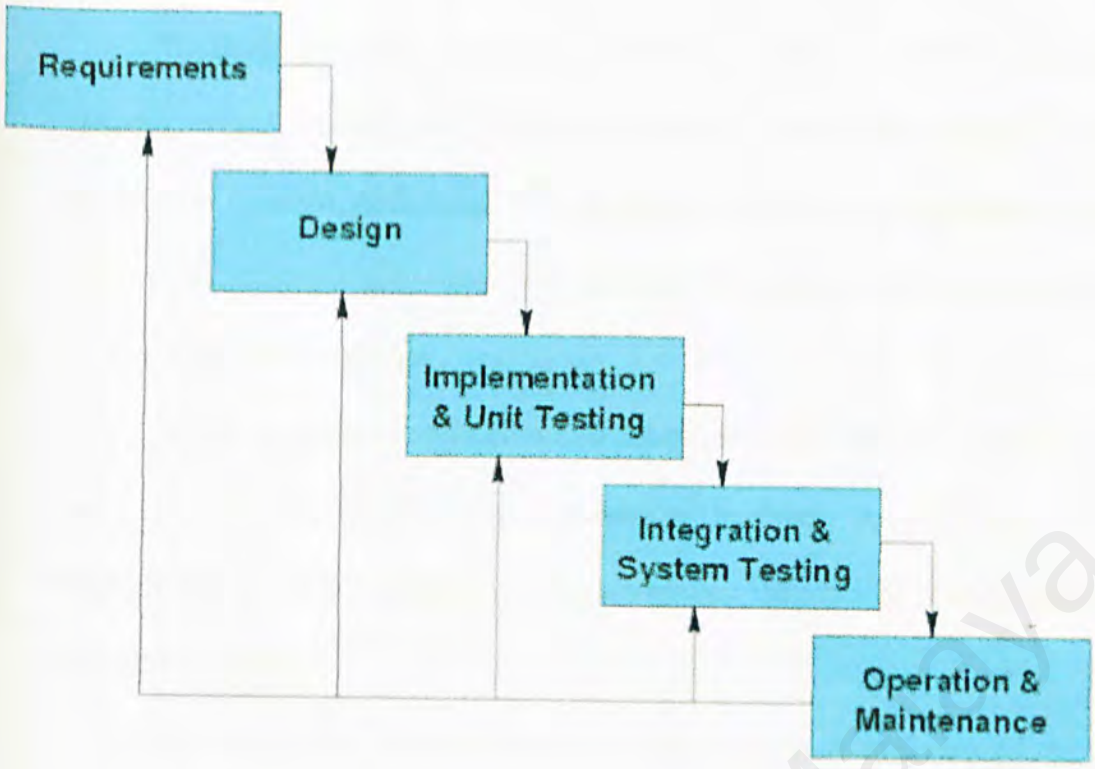


Figure 3.1: Project Development Methodology [16]

STAGE OF THE WATERFALL MODEL

1. ***Requirements analysis and definition.***

This stage is the requirements gathering stage and includes meetings and/or consultations with the user (system developers and the supervisor) to determine requirements of the system. These are usually the services it will provide, its constraints and the goals of the software. Once these are established they have to be defined in such a way that they are usable in the next stage. This stage is often precluded by a feasibility study or a feasibility study is included in this stage. It could be called the conception of a software product and might be seen as the very beginning of the life cycle.

To developing the Interactive 3D Driving Test Simulation, the requirements gathered would include the interaction between subsystem, business functionality, information retrieval, behaviour, performance, interface and constraints of the system. The requirements including functional and non-functional are documented and reviewed.

2. ***System and software design.***

In this stage the established requirements, flowing from the requirements analysis and definition, are identified as software or hardware requirements. The software requirements are then translated in such a way that they can be readily transformed into computer programs.

For this project, system design focuses on distinct attributes of the modules in Interactive 3D Driving Test Simulation program. The main page appearance and the organization of subsystem were designed in this stage. The overall system architecture, content design, interface representation, data structure, conceptual design and technical of this project are also required in this stage. In addition, the project system design also includes transforming the requirements into a representation that can be assessed for quality before the code generation (implementation stage) begins.

3. ***Implementation and unit testing.***

This is the stage where the computer programs are created. Each program is called a unit, and unit testing is the verification that every unit meets its specification.

During this stage, set of programs or program units section in Interactive3D Driving Test Simulation modules like Users Login Section will be create in the system based on the project system design. After code has been generated, unit testing is performed to verify that each unit meets its specification.

4. ***System testing.***

As the software is created and added to the developing system, testing is performed to ensure that it is working correctly and efficiently. Testing is generally focused on two areas which are internal efficiency and external effectiveness. The goal of external effectiveness testing is to verify that the software is functioning according to system design, and that it is performing all necessary functions or sub-functions. The goal of internal testing is to make sure that the computer code is efficient, standardized, and well documented. Testing can be a labor-intensive process, due to its iterative nature.

All the units are combined and now the whole is tested. When the combined programs are successfully tested the software product is finished.

This stage tests the complete of Interactive 3D Driving Test Simulation system. All the program units will be combine together (integrated) and tested as a complete system to verify that the software requirements have been met. The system is delivered to the user after the testing process.

5. ***Operation and maintenance.***

Maintenance takes place after the system is put into practical use. This stage is part of the life cycle of a software product, and not of the strict development, although improvements and fixes can still be considered as "development".

Maintenance involves correcting errors that have gone undetected before (in earlier stages of the life cycle), improving the system implementation, adding performance or functional enhancements or making changes due to accommodate changes in the software external environment for overall Interactive 3D Driving Test Simulation system.

3.22 JUSTIFICATION OF METHODOLOGY

The reasons for choosing Waterfall model as the development methodology are:

The waterfall model main strength is on its simplicity. Simplicity means that the development methodology is simple and easier to understand. Hence we can have a better and clearer guideline on what we shall do during development process.

With the adoption of the waterfall model, I just need to emphasize on one stage at one time and do not need to think of the next stage before the current stage is completed and therefore allowing me to have more focus and attention on what I am currently doing instead of over burdening myself with the upcoming stage.

By looking back at the model, we can say that although we have proceed to the next stage, if we found that there is something in the previous stage which has been left out we can still go back to the previous stage to rectify the problem. Such feature is important especially if the user at the development time wanted to add or remove certain functionality which other methodology cannot offer. So we can say that the waterfall methodology offers us a backward feature whereby we can rectify errors or incomplete of certain criteria in the previous stage although we have proceeded into the next stage.

The Waterfall model itself can be used for single projects that are cost or time restricted because of the easy way to manage it and cost and time restrictions are often an important issue. This methodology is more emphasize on planning rather than rapid development. It allows estimation of the completion of each stage so that the system can be developed within the time frame.

Simplicity is also essential when to present or explain or development progress to the user especially those who are not familiar with software development cycle. The model can give the user a better understanding on what is going on or the progress of the project development. If I am to adopt a more complex methodology, user may find it difficult to understand what I am trying to tell them.

3.3 INFORMATION GATHERING METHOD

Method of gathering information regarding a system is necessary in order to establish understanding of the state and future requirement on the system study and provide the groundwork for the system design.

There is no underlying standard or procedure to be followed strictly as each single project is unique and data-gathering may be vary to suit the needs of each particular project. However, there are a certain number of methods that are commonly used in gathering-information such as collecting hard data like written documents or reports, interviewing, using questionnaires, observation and sampling.

As for this project, due to cost and tight schedule constraints as well as the difficulties in finding and getting domain experts whom are willing to help, method such as questionnaires obtained the user's requirement from the user. The main data sources for system analysis were written documents, reference books, observation and other sources from the Internet.

I. Internet Sources

Internet is being used to seek information about the latest technologies, existing system that similar to interactive 3D Driving Test Simulation and information related to this project. Site visits and joining related newsgroups on the World Wide Web (www) are important to obtain a vast amount of up-to-date information from all around the world. Beside that, online tutorials regarding 3D Modeling and Engine development were obtained through surfing the internet. I also get a lot of information on system development, development tools and technologies, programming languages,

application programming interfaces, project methodology, and also more knowledge. The result from this research has been elaborated in detail in Chapter 2.

II. Book and References

Book and references are used to get the information that needed to develop the system. This including information from information system references, development tools references, programming references and database references.

Faculty of computer sciences documents room has a lot of senior's thesis that can be as a guideline for to write the thesis report. The format type of the report, organization of the heading and the content of the report can be referring from senior's thesis.

III. Monograph

A lot of published literatures were read in order to gather information of the user's needs. This is important especially to obtain system development needs and technical issues of the proposed system. All these can be categorized into the printed material (especially books and journals), and non-printed materials, such as electronic documents. Knowledge attained through reading are incorporated into ideas and implemented into the Interactive 3D Driving Test Simulation project. Useful information has been found in Chips, PC Magazine and In-Tech, which is published by a local newspaper, The star. This entire magazine provides the latest news about technology in the computing environment.

IV. Questionnaires

60 questionnaires form were passed out to student and staff of University Malaya. Subjects were asked to answer questions regarding to Interactive 3D Driving Test Simulation. The data obtained where then collected and analyzed.

V. Discussion with Supervisor

A discussion with supervisor has been practiced from time to time in order to get help and advices during the development of the project.

V. Knowledge Acquisition

This module is including this process to obtain and information, and an interview has been made to these people at two driving school:

- Puan Zalinah Ithnin, Clerk, Surfine Hightech Sdn Bhd (Kampung Kerinchi branch), Kuala Lumpur.
- Mohd Fatah, Manager, Sekolah Memandu Semasa, Bangsar Kuala Lumpur

Interview has been done to gain information from these people to complete the reviews of the existing system. Beside that this interview is to get information on the route that has been approved by Malaysia Ministry of Transport (Jabatan Pengangkutan Jalan) that used for the driving test.

3.31 QUESTIONAIRES ANALYSIS

There are three main parts in the questionnaire, Part A, Part B and Part C. Part A needs respondents to fill their personal particular, Part B is related to project information awareness. Whilst Part C is relate to driving education awareness. The results of the analyst are shown below:

{Please refer appendix for the sample questionnaires for the sampling process.}

PART A: PERSONAL PARTICULAR (QUESTION 1-4)

There were 60 respondents taken as a sample which involved majority of students in University Malaya and public. Below are the results of personal data survey.

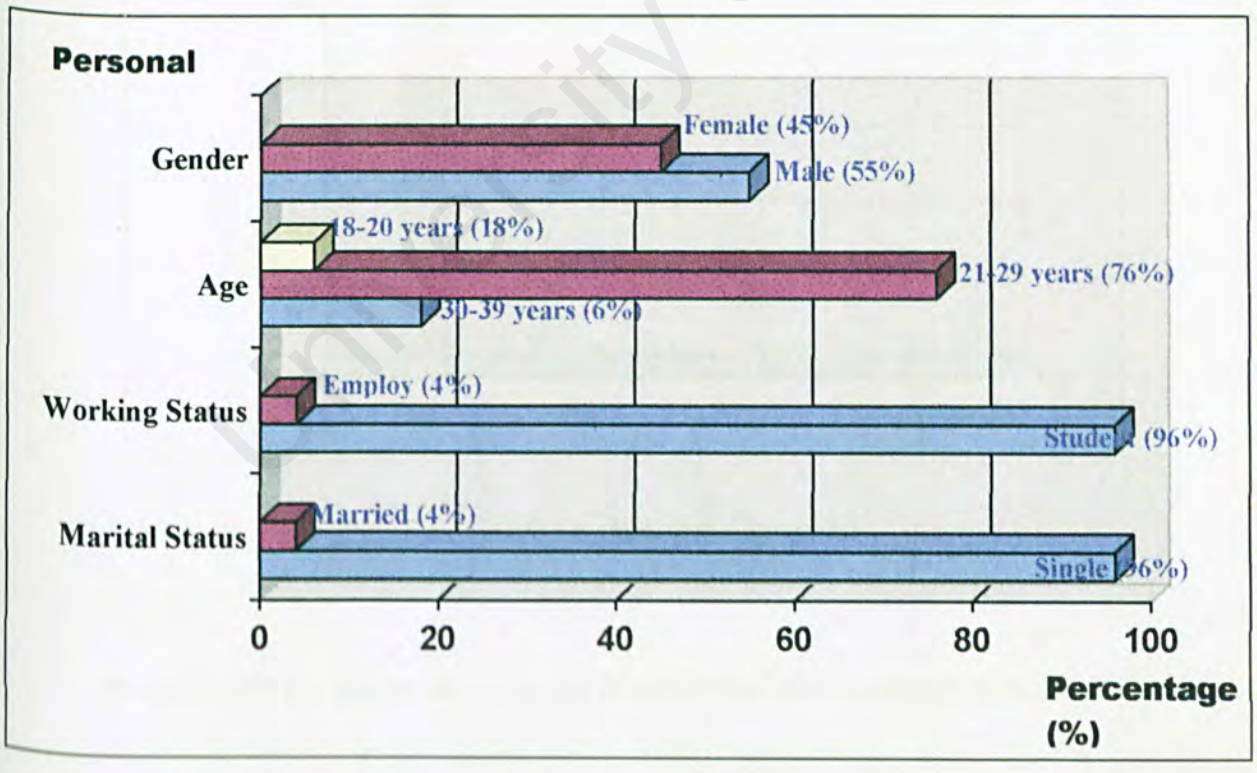


Figure: Percentage of respondents personal criteria

PART B: INFORMATION AWARENESS (Question 1-10)

There are ten question asked in Part B. Following are the results of the Part B Questionnaires – Information awareness.

Question 1

1. Do you have a personal computer at home?

This question is to analyze the number of individual who have a computer at home.

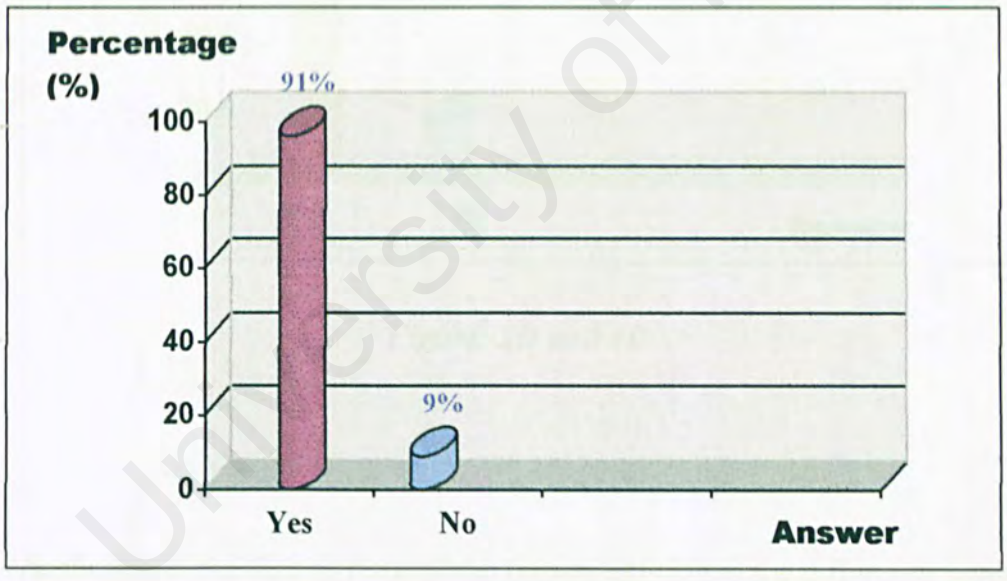


Figure: Personal Computer

According to the figure, the majority of individual own a computer at home.

Question 2

2. Do you know about 2D and 3D application? (eg: computer games)

The purpose of this question is to analyze the respondent's knowledge on the 2D and 3D application.

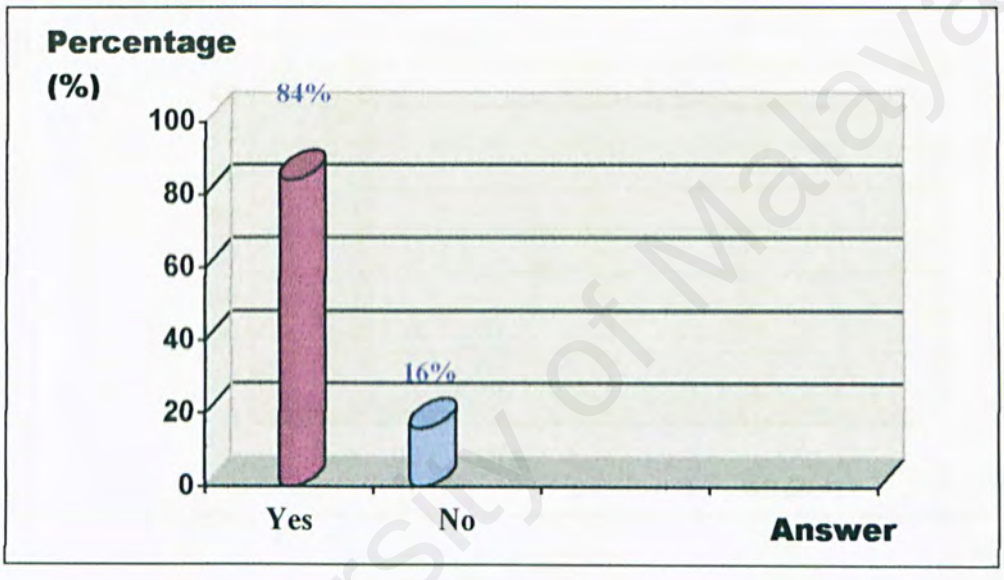


Figure: 2D and 3D

According to the figure, most of the respondents know the difference between 2D and 3D application.

Question 3

3. Have you used 2D application on computer or internet?

The purpose of this question is to determine the number of respondent that ever uses a 2D application.

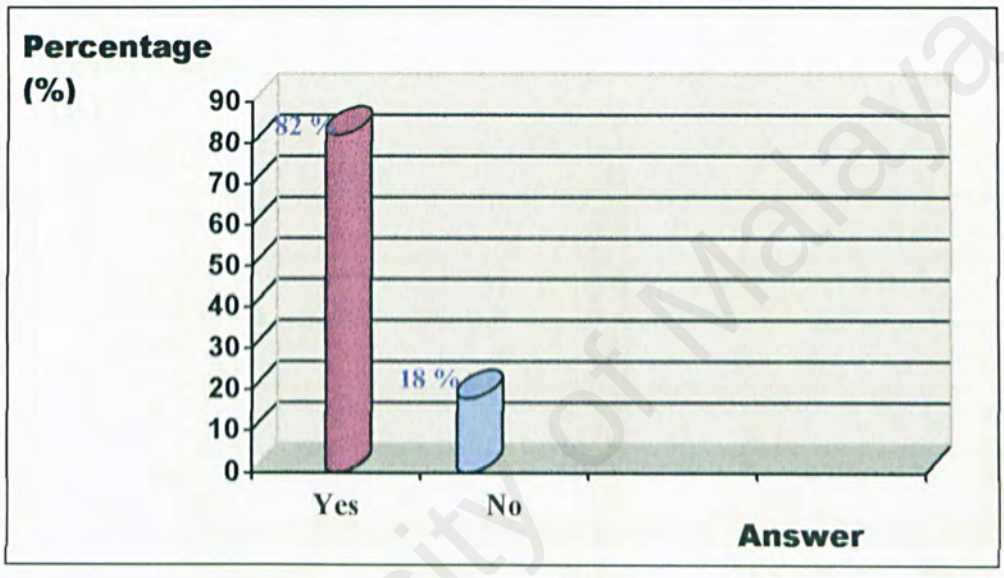


Figure: 2D Application

According to the figure, 82% of the respondents use 2D application out of 100%.

Question 4

4. Have you used 3D Application on computer or internet?

The purpose of this question is to determine the number of respondent that ever uses a 3D application and for what type of purpose they are using it.

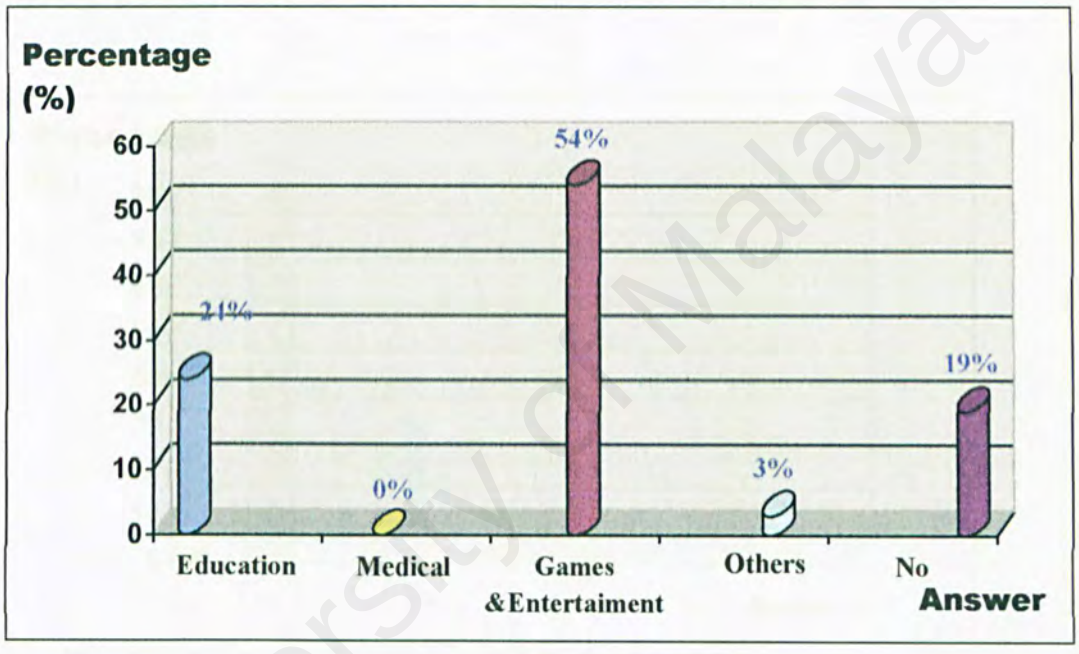


Figure: 3D Application

Figure, shows that 24% of respondent use 3D application for educational purposes, 54% for games and entertainment, 3% for other purposes. While 19% never use 3D application.

Question 5

5. Do you know differences between 2-Dimensional (2D) and 3-Dimensional (3D) in computer graphic and animation?

The purpose of this question is to analyze the respondent knowledge in 2D and 3D application, which related to computer graphic and animation.

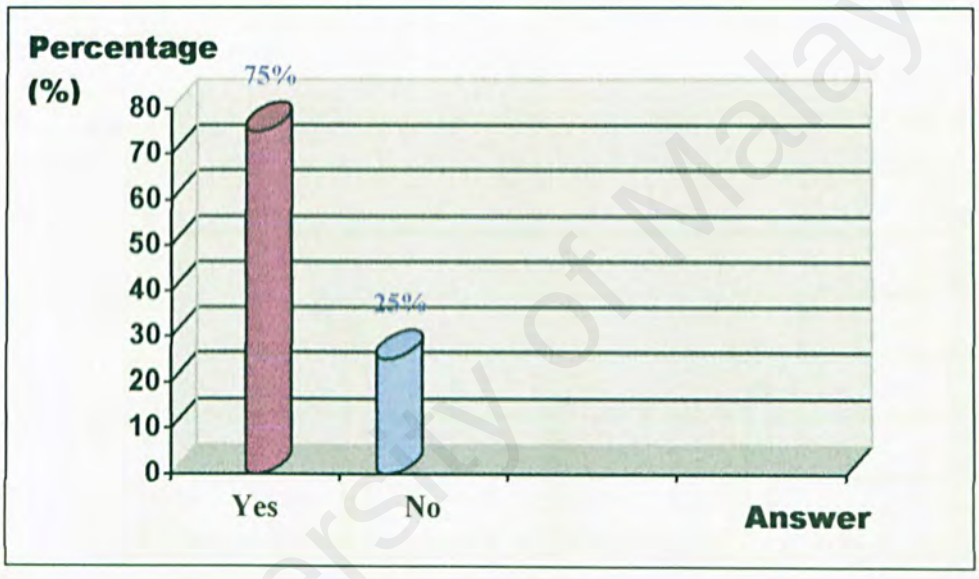


Figure: 3D and 2D Knowledge

The figure show that 75% of the respondent have the knowledge in 2D and 3D application, which related to computer graphic and animation

Question 6

6. Have you play 3D driving game before?

This question it to find out how much respondent enjoy playing driving game by analyzing the amount of time they spend on their computer daily to play driving game.

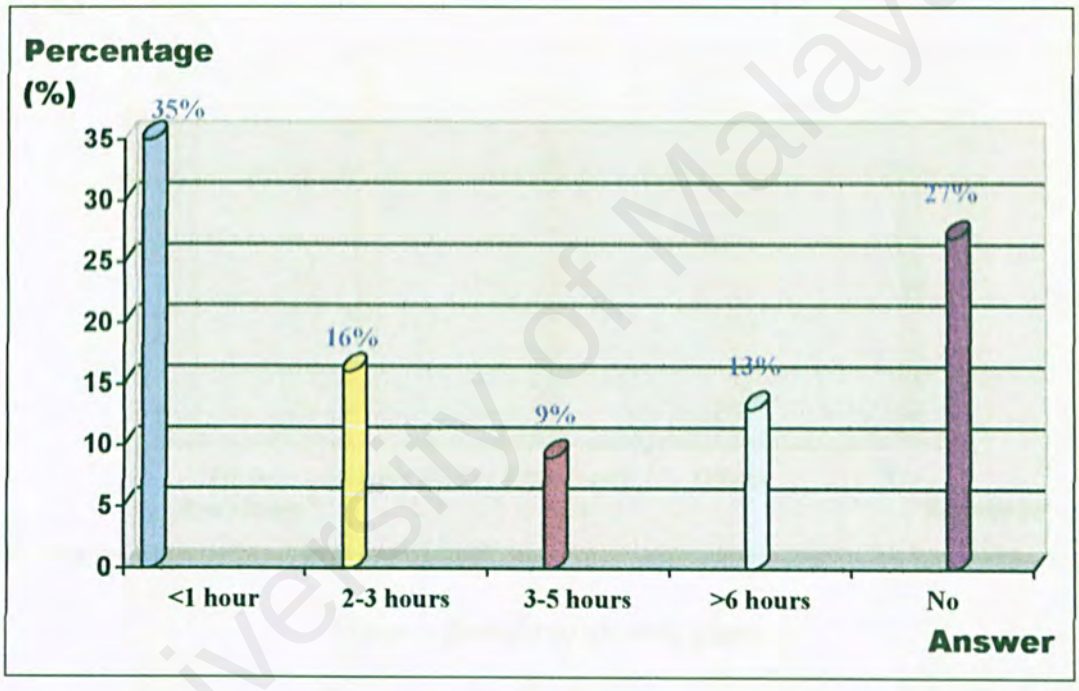


Figure: Times Spent on Driving game daily

According to the figure, respondent spent at least one hour on driving game per day, while some of them spent more than six hours daily on driving game.

Question 7

7. Do you think playing 3D driving games give you benefits?

The purpose of this question is to summarize the user opinion on driving game's benefit

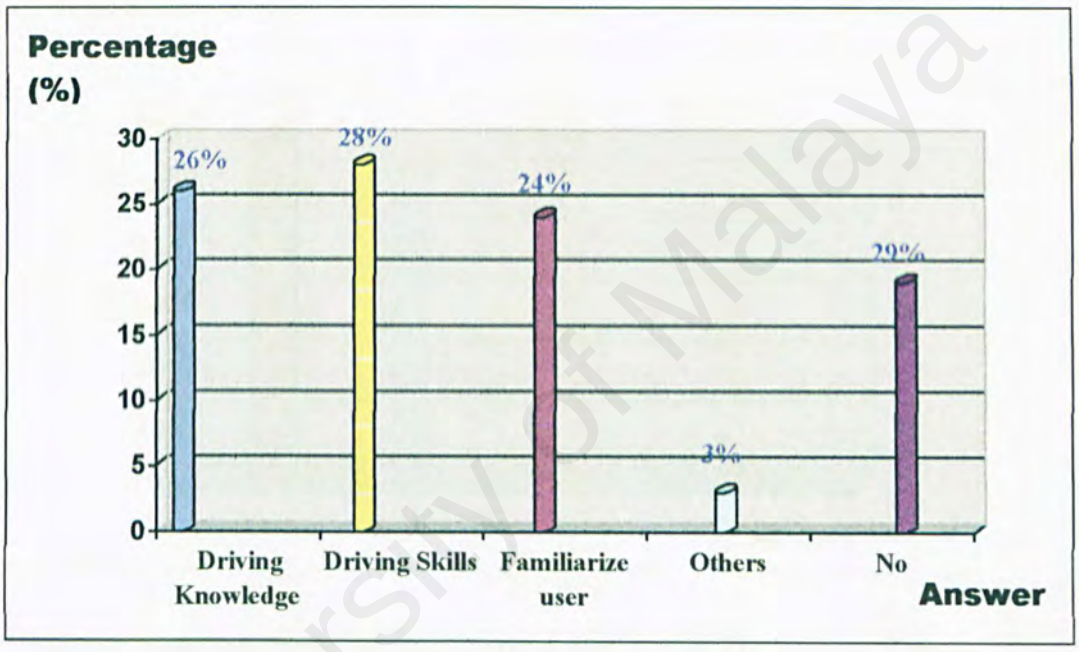


Figure: Benefit of driving game

According to the figure, most of the respondents think that driving game can improve driving skills. Some of them also agreed that driving game can deliver driving knowledge theoretically as well as familiarize user with real driving test situations.

Question 8

8. Do you think Interactive 3D Driving Test Simulation is similar to 3D driving game?

The purpose of this question to analyze users opinion o the similarity of the Interactive 3D Driving Test Simulation and 3D driving game.

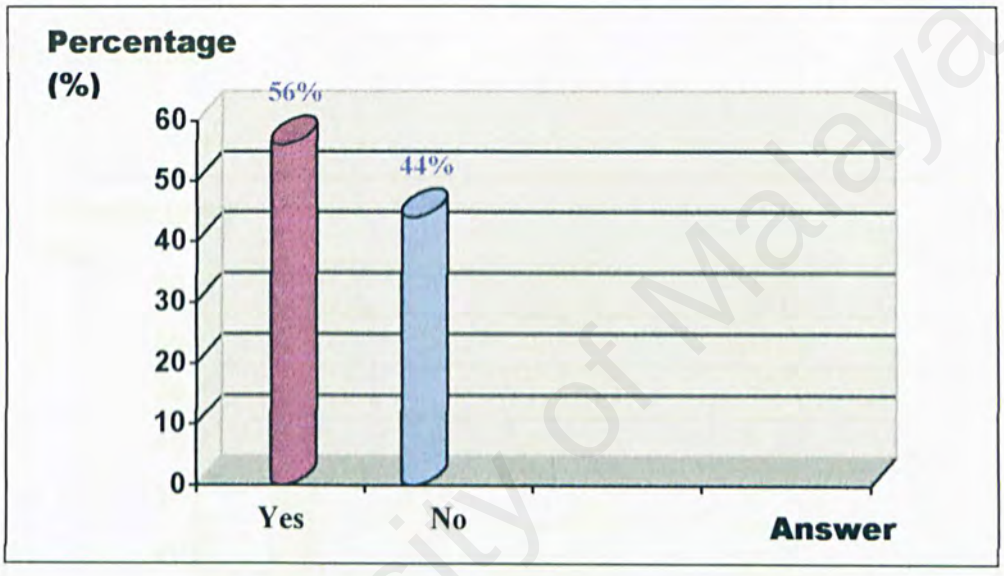


Figure: User's opinion on similarity of Interactive 3D Driving Test Simulation and 3D driving game

Figure shows that, 56% users agreed that Interactive 3D Driving Test Simulation is similar to 3D driving game, while 44% percent do not agreed.

Question 9

9. Interactive 3D Driving Test Simulation can be used to help user to familiarize them for the real time driving test environment. Do you agree with this statement?

The purpose of this question is to analyze respondent's opinion on the use of the Interactive 3D Driving Test Simulation to familiarize user with the real time driving test environment.

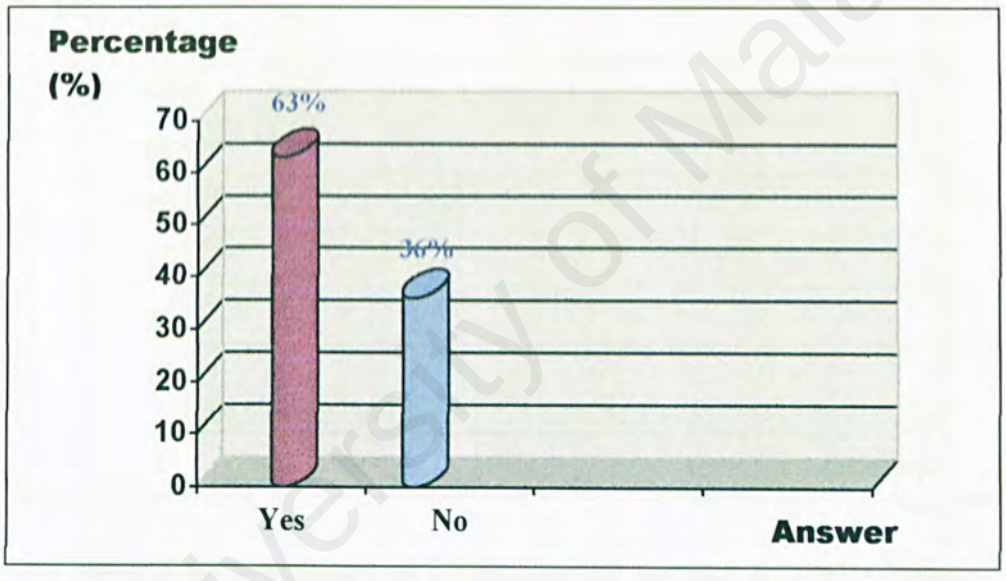


Figure: User's opinion on Interactive 3D driving Test Simulation

Figure shows that 63% agreed with the statement while 36% do not agreed.

Question 10

10. If the courseware of interactive 3D driving Test Simulation is develop, would you purchase the courseware?

This question is to analyze user's acceptance for interactive 3D driving Test Simulation courseware.

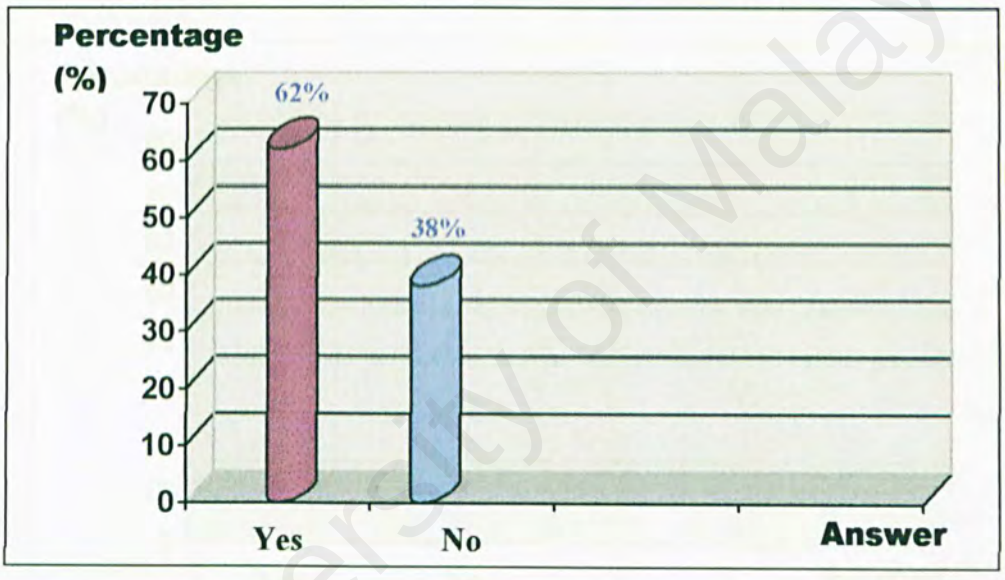


Figure: User acceptance of Interactive 3D Driving Test Simulation

Figure shows that 62% out of 60 respondents would buy interactive 3D driving Test Simulation courseware, while 38% would not buy the courseware.

PART C: DRIVING EDUCATION AWARENESS (Questions 1-3)

This part consists of three questions. The objective of Part C is to gain information on respondent driving education awareness.

Question 1

1. Do you have a driving license?

This question will analyze the number of respondent posses a driving license.

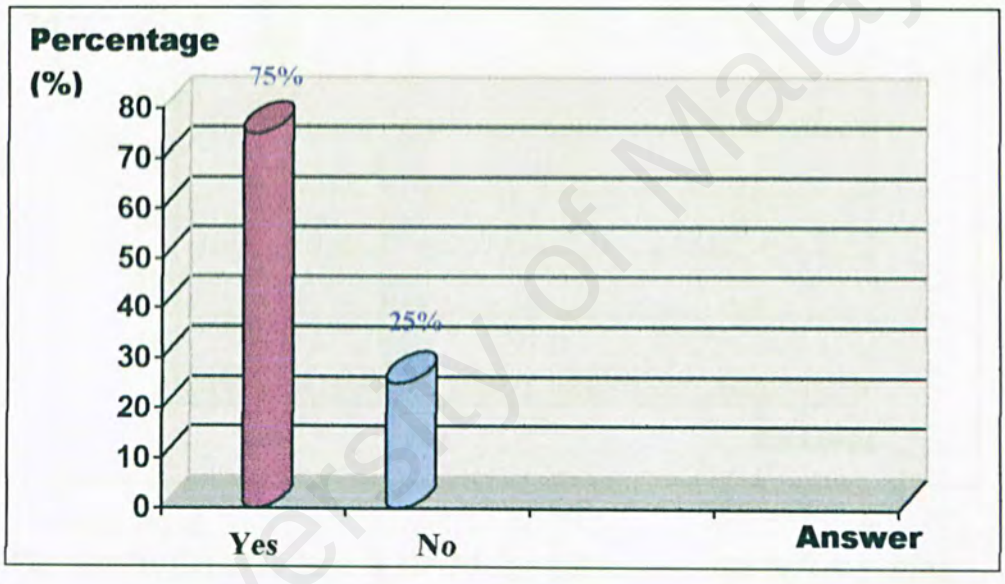


Figure: Driving License

According to figure, majority of the respondent posses a driving license.

Question 2

1. Do you prefer to have a 3D driving test courseware before you are to get a driving test license?

The purpose of this question is to analyze whether the respondent prefer to have 3D driving test courseware before getting a driving test license.

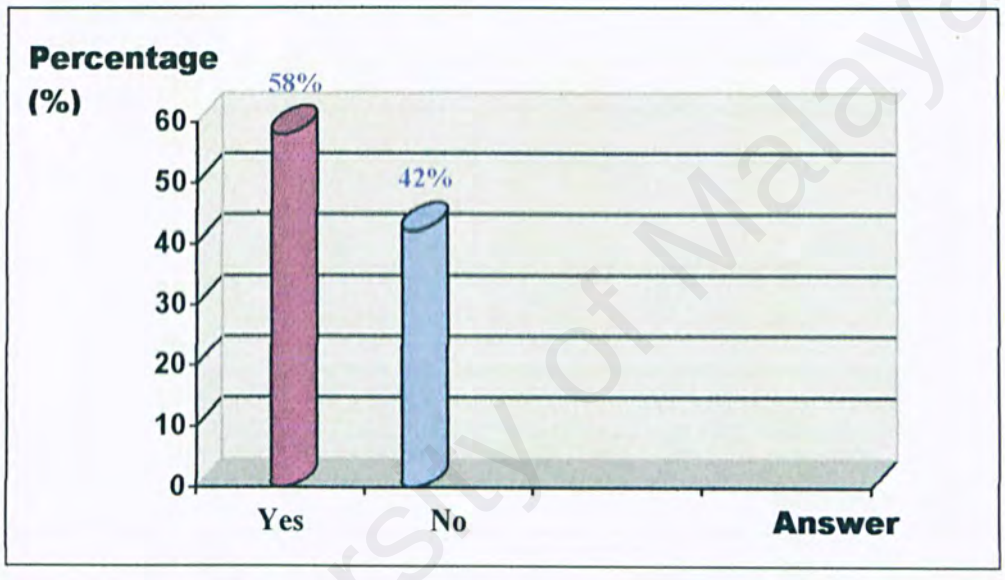


Figure: Preferred to have a 3D driving test courseware before getting a driving test license

According to the figure, 58% of respondents prefer to have 3D driving test courseware before getting a driving test license, while 42% decline to have it.

Question 3

2. Do you think Interactive Driving Test simulation can help you to improve your driving skills?

The purpose of this question is to find out whether the respondent agreed that Interactive Driving Test simulation could help to improve their driving skills.

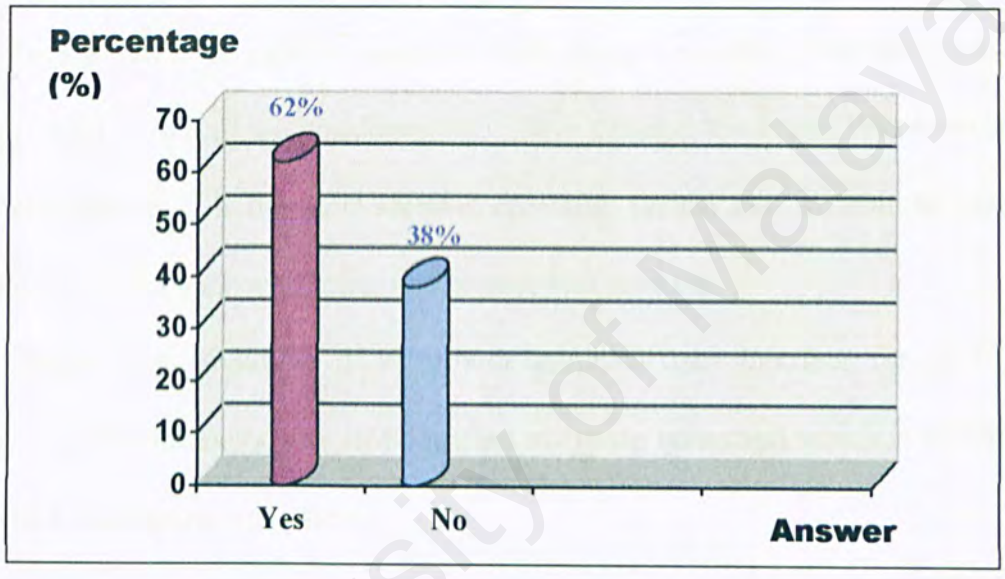


Figure: Interactive Driving Test Simulation can help to improve real driving test skills

According to figure, 62% agreed that Interactive Driving Test simulation could help to improve their driving skills.

3.4 CONCLUTION ON TECHNOLOGY AND TOOLS ANALYSIS

After all the technologies have been reviewed and analyzed, the most suitable and appropriate tools for developing the system are identified and selected. The tools to be selected include the development software as well as the entire platform on which the development of the project is occurred.

Consideration of Development Platform

Review has been done in purpose of choosing a suitable platform/operating system for this project. After all the consideration, I have decided to choose **Windows 2000** as my system platform, as it is the most suitable operating system that are able to support all the development tools I choose to bring this project into success.

Besides that, Windows 2000 provide graphical user interface for user friendliness whereby user just interacts with icons instead of typing command which is common in DOS and UNIX based operating system.

Advantages of Windows 2000:

- An enhancement of Windows NT 4.0.
- It is suitable for enterprise or organizational level.
- Showing high performance, reliable, secure and easy-to-manage characteristics for information sharing and running applications.

- The Task Manager of Windows 2000 is an integrated tool for monitoring applications and reports key performance metrics of the Windows 2000 system. It provides information on each application and process that are running on the workstation, as well as memory and CPU usage.
- System Policy Editor and User Profiles of Windows 2000 allow system administrators to manage and maintain users' desktops in a consistency manor. System policies are used for the standardization of desktop configurations and control the user work environment and actions.
- New application programming interfaces for server application developers and better server performance deliver improved throughput and scalability for server applications.
- Improved Windows 2000 Diagnostics Tool allowed for easy examination of the system resources such as IRQ, DMA and IO addresses, all presented in an easy-to-view graphical tool.
- Windows 2000 allows Object Linking and Embedding (OLE). It can combine the information from several applications into one compound document using the special OLE capabilities of Window-based application.

Consideration of Development Tools

Selected Application Programming Language

For this project, Interactive 3D Driving Test Simulation is designed with Maya as an application programming language for modeling. Since the programming language must engage to the Application Programming Language (API), Maya is more suitable API's apply in the development of this system to perform the better quality and attractive modeling picture.

Advantages of Maya:

- A new hardware rendering option takes advantage of the power of next-generation graphics cards to produce near software-quality images at often dramatically faster speed.
- A unified rendering workflow provides easy and consistent access to the Maya rendered through a common interface.
- Animation enhancements to constraints, forward/inverse kinematics and ghosting give animators new levels of flexibility.
- Maya Paint Effects(TM) to polygon conversion offers a whole new range of looks, as well as more editing and output options.
- An improved polygon reduction method is valuable to users making lightweight game models and MODs.

Consideration of Graphic library

Selected Programming Language

Since the graphic library must be suitable with Application Programming Language (API), I have selected combination of OpenGL and C++ to use in my project. It is because the combination of OpenGL and C++ can support real time rendering. OpenGL is used widely in CAD/CAM, games and other high end applications. The combination of these programming languages can give the system fast simulation rather than others programming language.

Advantages of OpenGL:

- **OpenGL delivers fast and complete 3D hardware acceleration.**

  Manipulate massive amounts of data in real-time by using OpenGL hardware accelerated geometry, real-time lighting, clipping, transformations and rendering

- **OpenGL makes real-time 3D effects possible**

  OpenGL hardware acceleration adds detail & special effects to images without compromising performance. Examples include real-time fog, anti-aliasing, volume shadows, bump mapping, motion blur, transparency, reflections, 3D textures, volume rendering and more

- **OpenGL is designed to support future innovations in software and hardware**

  The extension mechanism in OpenGL can expose radically new hardware/software features that simply never existed when OpenGL was originally conceived. Users will the best possible performance from applications and games as hardware technology advances

- **OpenGL is on every platform**

  Ever wished that great Unix application would move to Windows or a great Windows game to Linux. Since OpenGL is fully cross-platform, it is relatively easy to port games or applications from one platform to be moved to another. This greatly expands the wealth of applications and games that support OpenGL.

- **OpenGL is stable.**

  High End 3D workstation & supercomputers users have been enjoying OpenGL since 1992. Today, OpenGL is also the most popular and powerful 3D solution for Windows, Mac and Linux-based computers, for professional and consumer use.

Advantages of C++:

- C++ is a better C
- Supports data abstraction
- Supports object-oriented programming
- Supports generic programming.

3.5 CHAPTER SUMMARY

In order to produce a more efficient and better quality system, this project will be develop through several stages, by implementing the Waterfall Model methodology. The first two stages, which are requirement analysis and system design have been completed in the first semester. While for system coding, system testing, and system operation, will be complete in the second semester.

The waterfall model approach is selected for the development of Interactive 3D Driving Test Simulation is because a web-based a pc-based system needs a well organized and structured planning of system design and implementation. This method is adopted because simple and ease of implementing. In addition, testing is performed in every stage and may go back to the previous stage to correct any errors.

Through the system development life cycle, system methodology is adopted to understand the current problem situation. Careful analysis and research such as internet sources, questionnaires, monograph and others has been conducted to determine the feasibility of the system and what is required of it. The chosen development technologies are Maya, OpenGL and C++ and in other words, all the technologies that are chosen are the products from Graphic Corporation. The system requirements are identified, translated into design and finally implemented via coding. The finished system is evaluated to meet the system objectives and requirements specification.

CHAPTER 4

SYSTEM ANALYSIS

4.1 INTRODUCTION

A requirement [28] is a feature of the system or a description of something the system is capable of doing in order to fulfill the system's purpose. Capturing requirements means to understand what the users expect the system to do. Thus, our understanding of the system intent and function starts with an examination of requirement. Requirement should be documented and reviewed for correctness and completeness in a requirements review after they have been defined. Requirements are of two types, which are functional and non-functional requirements.

4.2 FUNCTIONAL REQUIREMENTS

A functional requirement [17] describes an interaction between the system and its environment. Further, functional requirement describe how the system should behave given certain stimuli module.

The system is divided into two modules which are:

- ***Route Module:*** This module gives a chance to user to choose a route.
- ***Help Module:*** This module list out the entire control button used in this system. It explains to user the usage of each control button.

Route Module:

Route module is including two routes that the user is able to choose whether they want to use Route A or Route B. Each route provides the different route but user has to pass away the same area from the different direction.

4.3 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements [21] describe a restriction on the system that limits our choices for construction a solution to the problem. These constraints usually narrow our selection of language, platform or implementation techniques or tools. The non-functional requirements of this application are described in the following sub- sections.

User Friendly

User should be able to browse and use the application with ease. Adequate help must be provided. This will require less time in learning to use this application.

Response Time

Response time to retrieve the screen must be within a reasonable interval time.

Efficiency

User is able to invoke an unlimited number of tomes and the system should be able to produce the similar results and outcome at a reasonable speed.

Portability

The system should be able to be used in many platforms.

Efficiency

User should be able to invoke an unlimited number of times and the system should be able to produces the similar results and outcome at a reasonable speed.

Maintainability

The system should be able to be corrected if an error is encountered, and adapted if the environment changes or enhanced if the user desires a change on requirements with ease. This system is an integration of a few modules with each respective function to enable any modifications being done towards each respective module without modifying other modules of the system.

Expandability

Architecture components, algorithm, data structure and procedures design should be able to be extended and modified with ease. This is important so that any future enhancements and expansion can be done easily.

Simplicity and attractiveness

Simplicity in the web application refers to keeping forms and screen properly uncluttered in a manner that focuses the user's attention. Simplicity promotes efficiency of the web application as well. Attractiveness refers to the user's enjoyment or attraction to use the system due to their appealing design.

Property	Measure
Speed	Processed transactions/second User/Event response time Screen refresh time
Size	K bytes Number of RAM chips
Ease of Use	Training time Number of help frames
Portability	Percentage of target-dependent statements Number of target systems

Table 4.11: Metrics for specifying non-functional requirements

4.4 SYSTEM REQUIREMENTS

Consideration of system Requirements

The hardware and software requirement are listed below:

Software Requirements for developers:

- Microsoft Window 2000.
- Microsoft Office – MS Word, Power Point.
- Microsoft Project 2000.
- Visual Studio 6.0.
- Visuals API
- Maya 4.0

Hardware Requirements:

The hardware requirements will be divided into two aspects, which are:

- System requirements for developer
- System requirements for user

System Requirements for Developer

The hardware and computer system that are needed to develop the Interactive 3D Driving Test Simulation should have a large hard drive, fast processor and a huge RAM because graphics require a large memory. Hardware such as CD-RW and scanner is also essential.

Main hardware for developers (minimum requirements):

- 3D Video Card or equivalent (64MB GeForce 4)
- Mouse
- 20GB hard disk
- Pentium 200MHz Processor or equivalent
- 128 MB Random Access Memory (RAM)
- CD-ROM drive
- CD-RW drive
- Sound Card
- Speakers
- DirectX 7.0 plug-in
- OpenGL library

System Requirements for User

A better system should have a good graphic card so the user can enjoy the route together with the sound provided. This application does not require any plug-in and it can be used directly from CD drive, without installing it to the computer.

User's main system requirements:

- Intel Pentium 200MHz Processor or equivalent
- Windows 98/NT/2000/XP
- 64 MB Random Access Memory (RAM)
- CD-ROM drive
- 640 x 480, 256 color display
- Sound Card
- Speakers
- Keyboard
- Mouse
- CD-ROM drive

CHAPTER 5

SYSTEM DESIGN

INTODUCTION

System design [28] is the creative process of transforming the problem into a solution. It is used to design, and implement improvements in the functioning of business that can be accomplished through the used of computerized information systems. We use the requirement [21] specification to define the problem. Then, we declare something to be a solution to a problem if it satisfies all the requirements in the specification. In many cases, the number of possible solutions is limitless.

5.1 SYSTEM DESIGN

Software design is a process of devising and documenting the overall architecture for a software system. It includes identifying the major components of the system, specifying what they are to accomplish, establish the interface among components. Design is the first step in the process of transforming the requirements into a close representation of the eventual functional software (Ow and Mashkuri, 1998).

Definition of design can be uttered in many ways. We can say that, the design is the create process of transforming the problem into a solution and the description of a solution is called design. All of the design must satisfy all of the requirement specification.

To achieve a good design and to make sure the quality of the software been taken very well, the design should:

- Carry out all the requirements which has in the analysis made & it must fulfill all the requirements desired.
- Be a simple and easy guidance to the programmers.
- Provide some sort of for the programmers.

Every specifications of the design describe the features of the system and every parts of the system, which will be appearance to users. This chapter discusses the system functionality design and user interface design of the Interactive 3D Driving Test Simulation.

5.2 SYSTEM FUNCTIONALITY DESIGN.

The arrangements of the structure chart shows tell us the relation between modules in this project. It is used to show the activities that make up the system. It gives the meaning that the requirements into system functionality can be used. The main thing of this design is it focuses on the system architectural design and user interface design.

Architectural design maybe based on a particular architectural model or style (Garlan and Shaw, 1993). Different designers approach the architecture design process in different ways. There are the common activities in architectural design process:

- **System structuring**

  The system is structured into a number of principle sub-systems are identified.

  Communications between sub-systems are identified.

- **Control Modeling**

  A general model of the control relationships between the parts of the system is established.

- **Modular Decomposition**

  Each identified sub-system is decomposed into modules and its interconnections.

Classification of decomposition methods:

- Function-oriented decomposition ([Wirth 1971], [Yourdon 1979])
- Data-oriented decomposition. [Jackson 1975]
- Object-oriented decomposition. [Rumbaugh 1991]

Structure Charts using System Structuring and Modular Decomposition

Structure charts are used for procedural programs to illustrate the following information about a program in a visual format:

- Partitioning of a program into named modules (functions)
- Top-down hierarchy and organization of modules
- Linkages between modules
- Flow of data, control or exception information

The figure below shows the main system structure chart of the Interactive 3D Driving Test Simulation system.

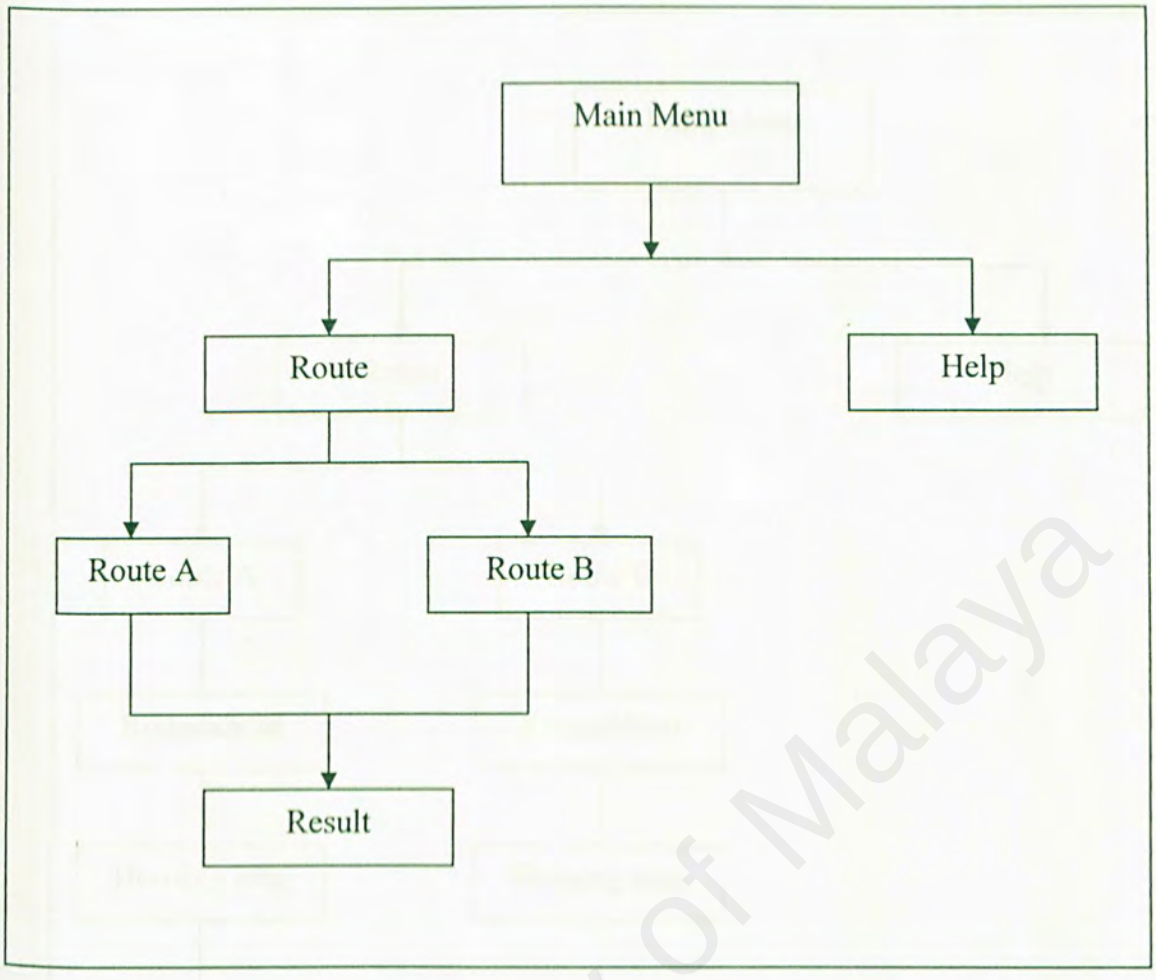


Figure 5.21: Main system structure chart (Main Menu)

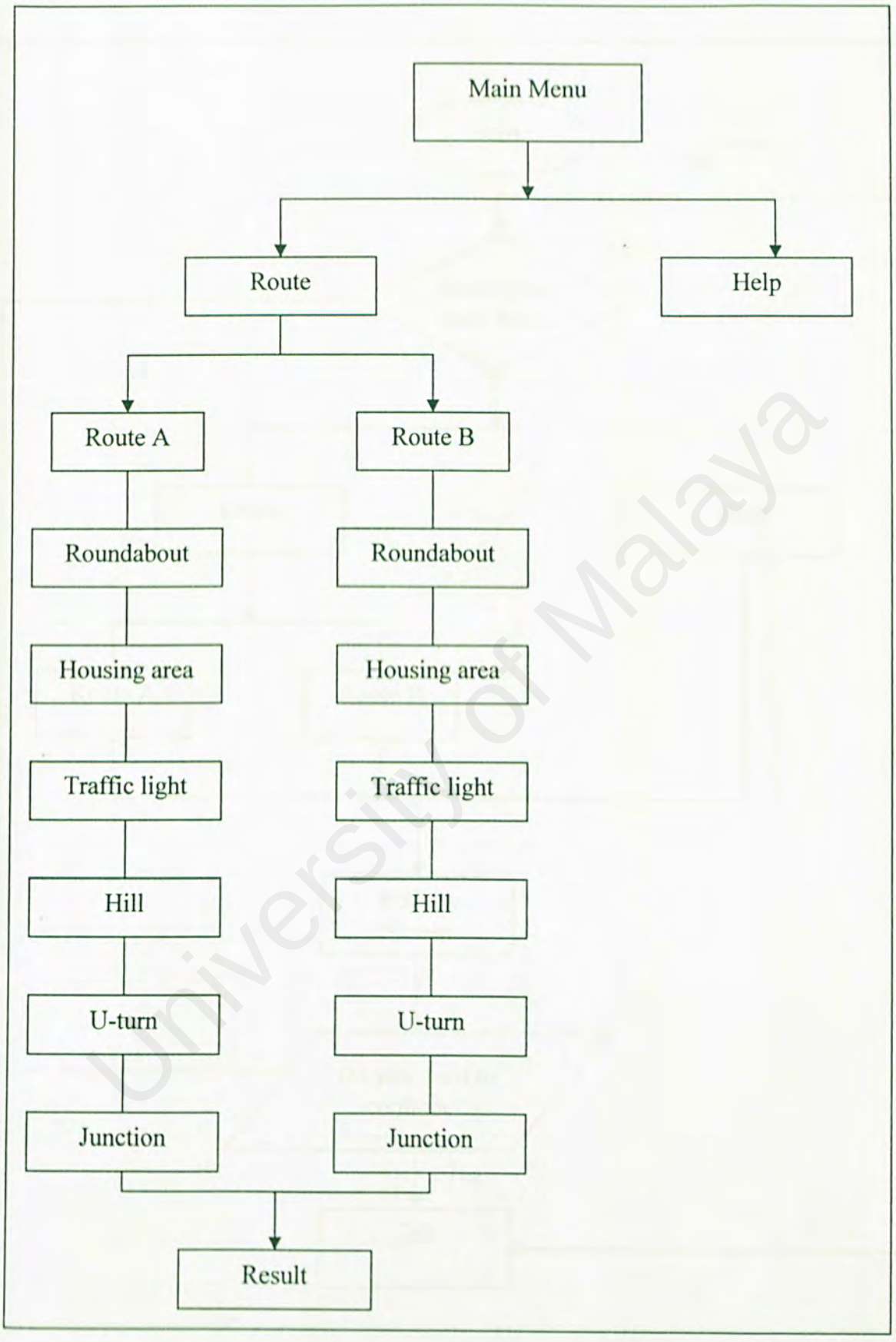


Figure 5.22: Structure chart of Interactive 3D Driving Test Simulation

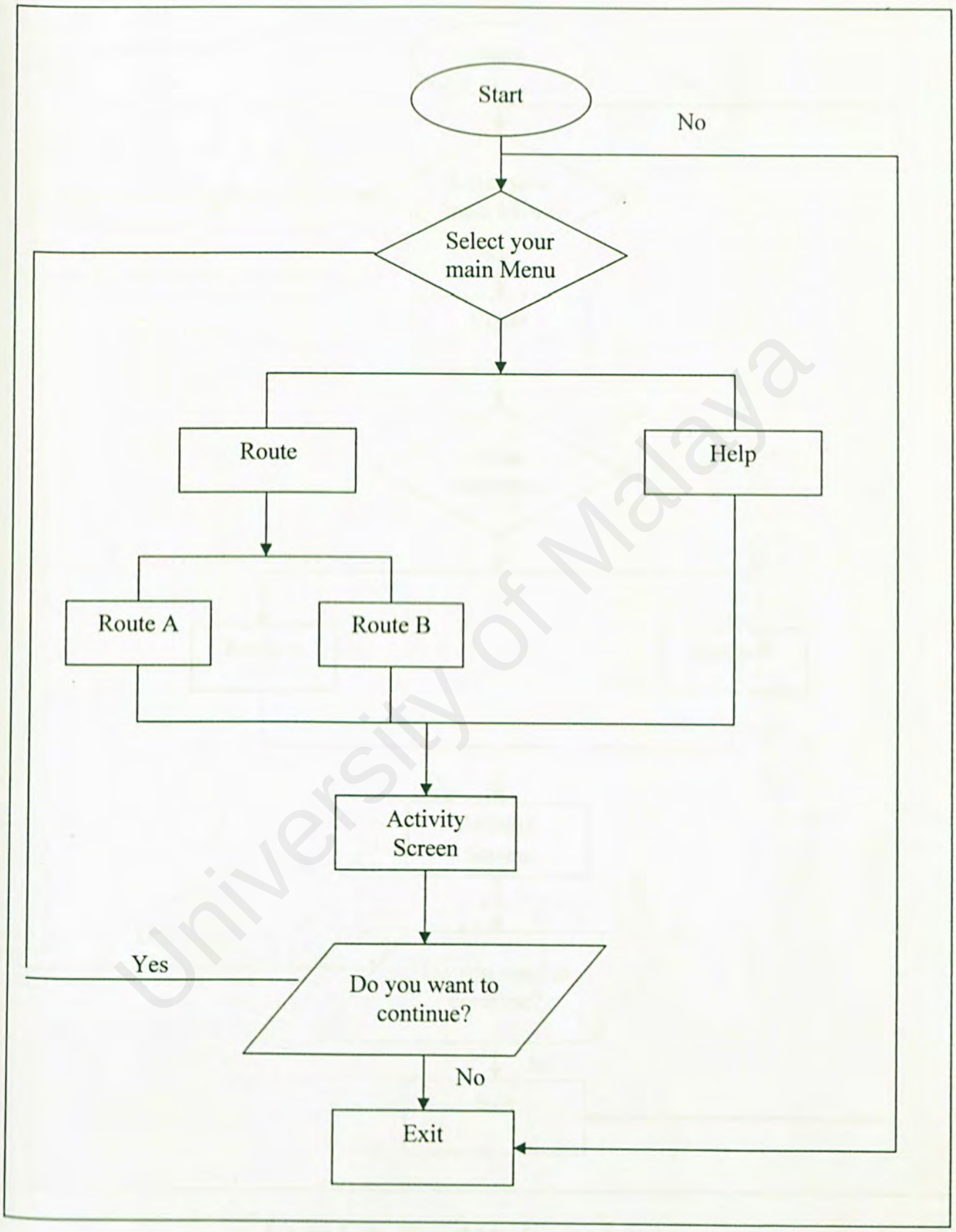


Figure 5.23: Flow chart of main menu

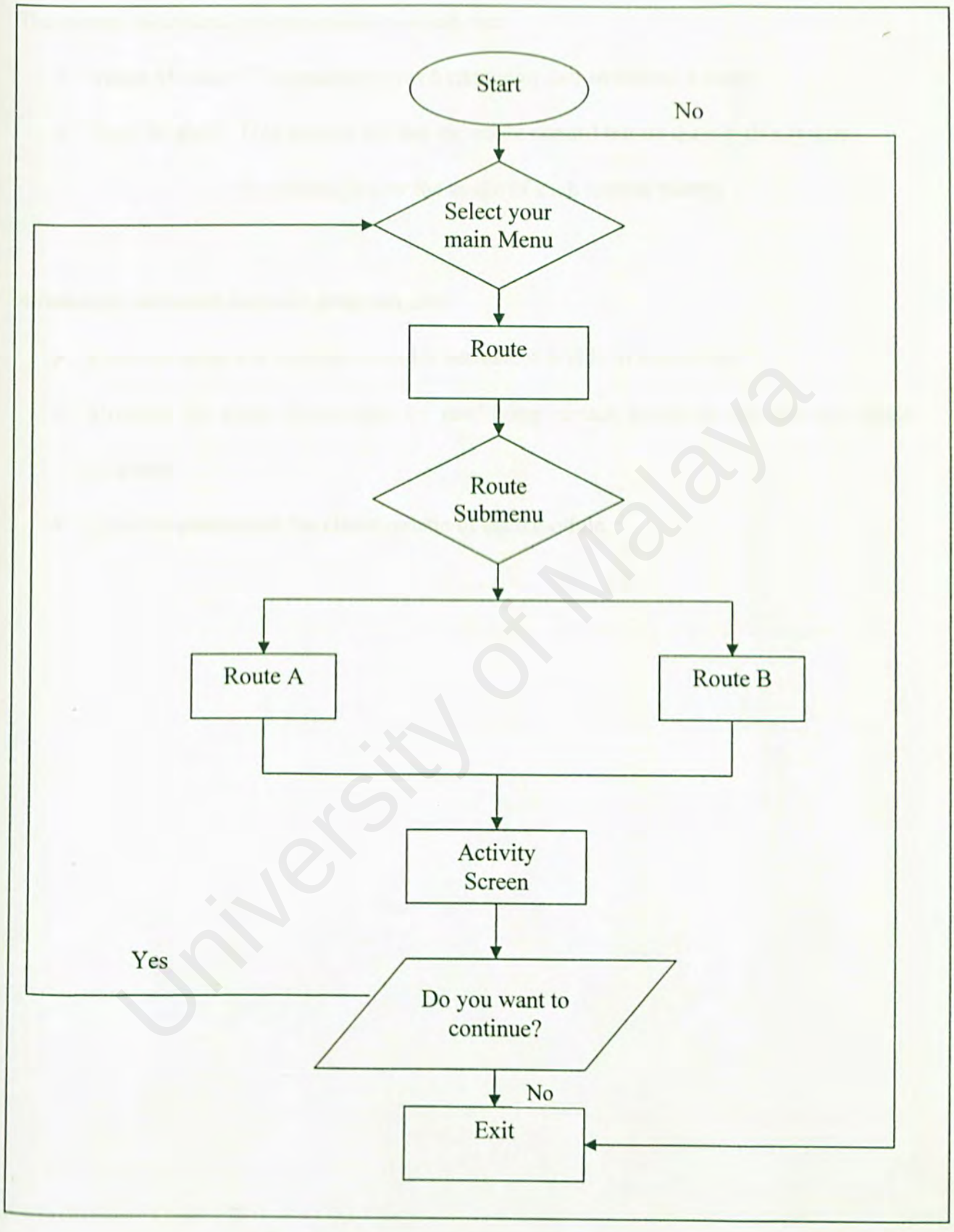


Figure 5.24: Flow chart of route menu

The system is divided into two modules which are:

- ***Route Module:*** This module gives a chance to user to choose a route.
- ***Help Module:*** This module list out the entire control button used in this system. It explains to user the usage of each control button.

Advantages on using modular program are:

- Easier to write and compile modules because it is able to standalone
- Modules are easier to manage for modifying certain functions and not the whole program.
- Easier to understand the characteristic of each module.

5.3 SYSTEM CONFIGURATION

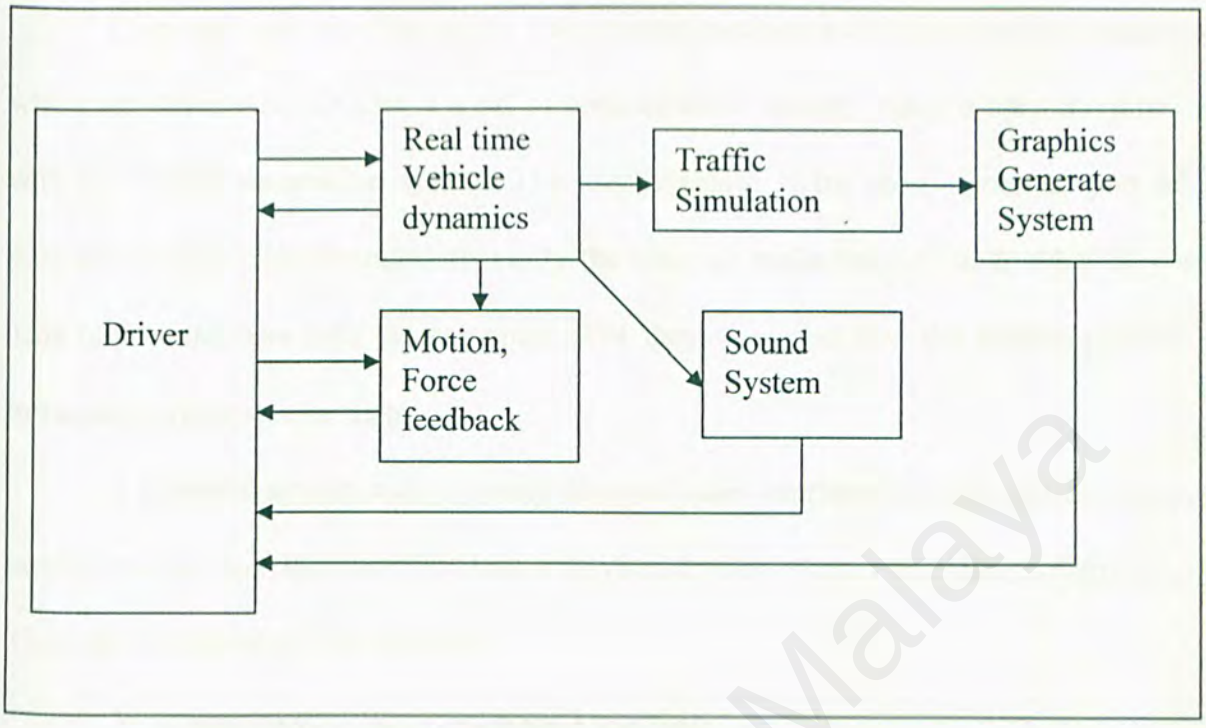


Figure 5.31: Interactive 3D Driving Test Simulation's system configuration

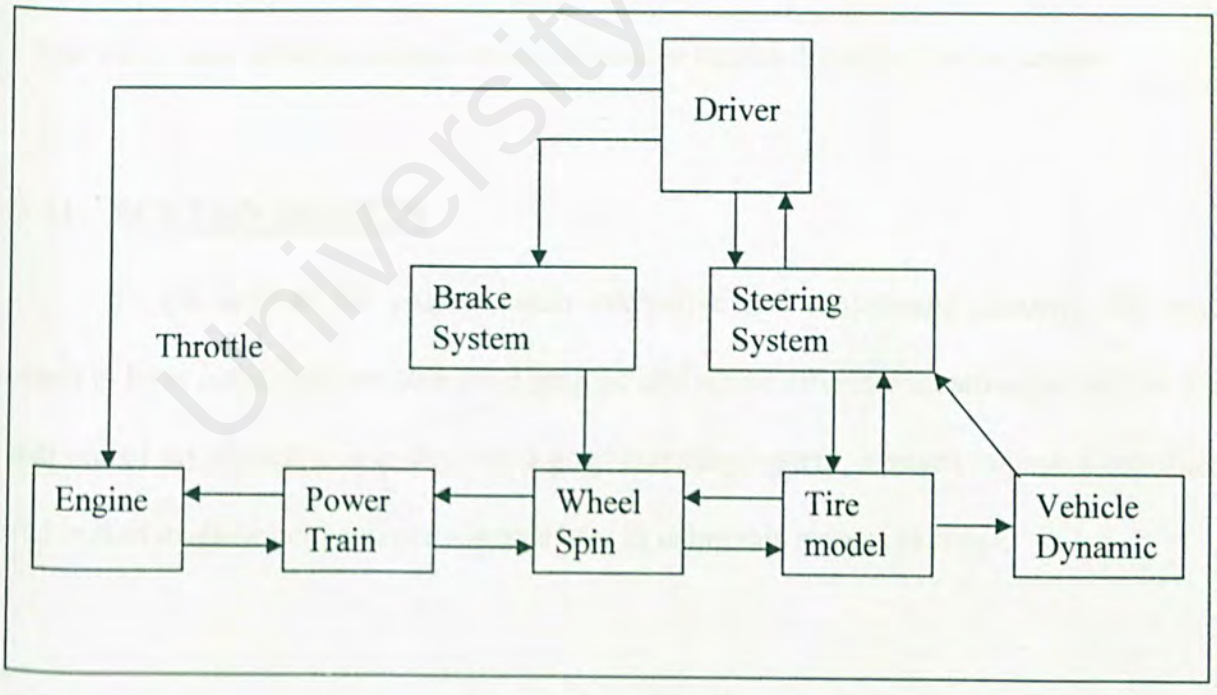


Figure 5.32: Real Time Vehicle Models

5.4 USER INTERFACE DESIGN

Computer user interface is the place where machine and human being communicate with each other. An interface is a set of commands or menus, which a user communicates with the library automation system. The user interface is the most important part of any program because it determines how easily the user can make the system do what they want. This is how the user tells the computer what they want and how the system presents the information requested by them.

A powerful system with a poorly designed user interface has little value. Today, the most common user interface involves a keyboard, view screen and more often a mouse. There are two types of user interface

- A command-driven (text based) interface
- A menu-driven (graphical user) interface

A command-driven interface is one user enter commands. A menu-driven interface is one which user select command from the various menus displayed on the screen.

5.41 SCREEN DESIGN

To get used to the package with interactive and multimedia concept, this system needs to have icons, buttons plus good graphic and sound effects with attractive colors. Since that one of my objective is to develop a good and easy system, it needs to have a lot of icons and button to allow convenience towards user in using this system package.

My screen design is shown below:

Screen design of main module:

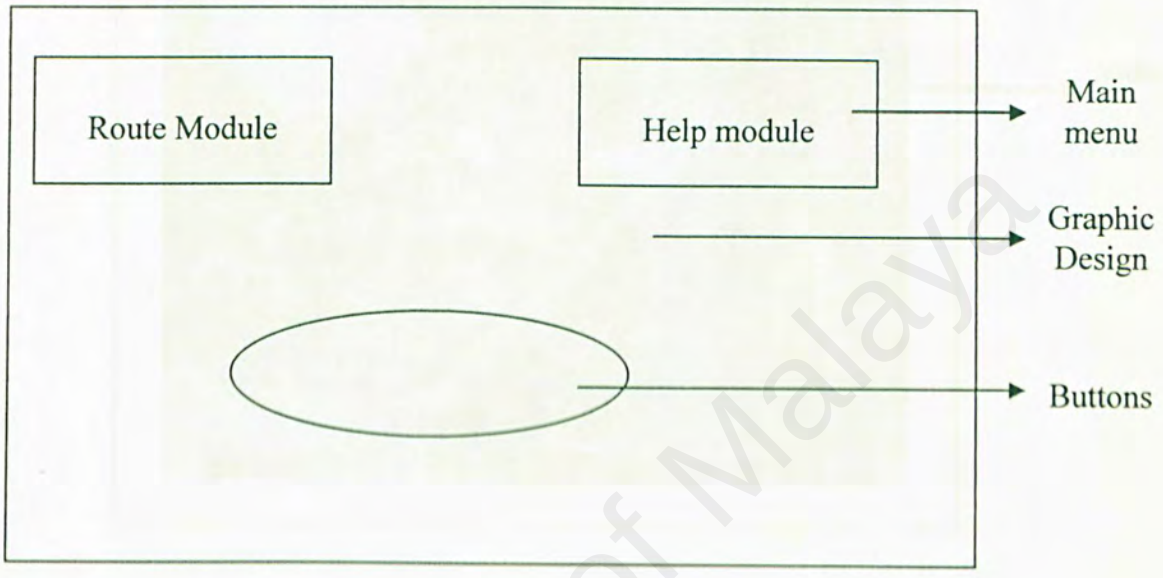


Figure 5.41: Screen design main menu

Screen Design of Route Module:

Basically, this will be the design for all screens in the multimedia training package. Below is the example of screen design on route module.

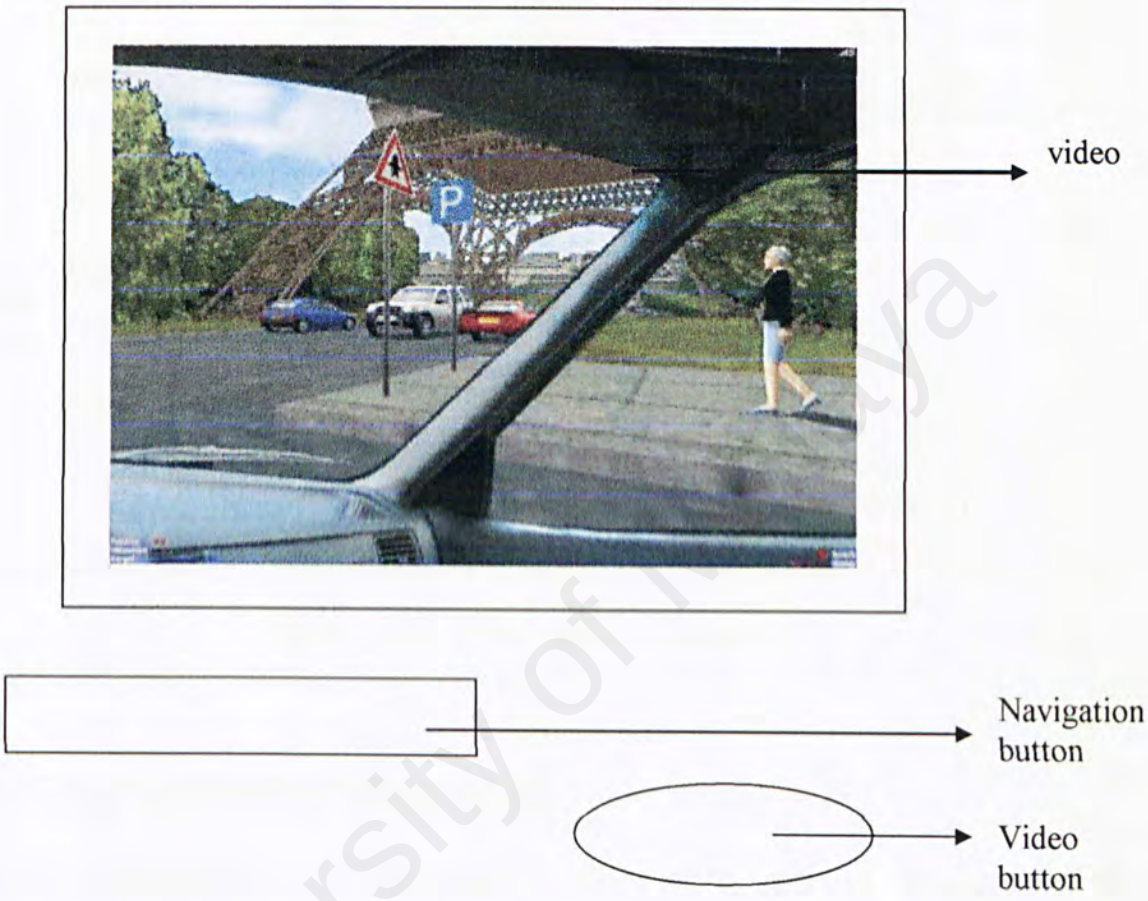


Figure 5.42: Example Screen Design for route module [25]

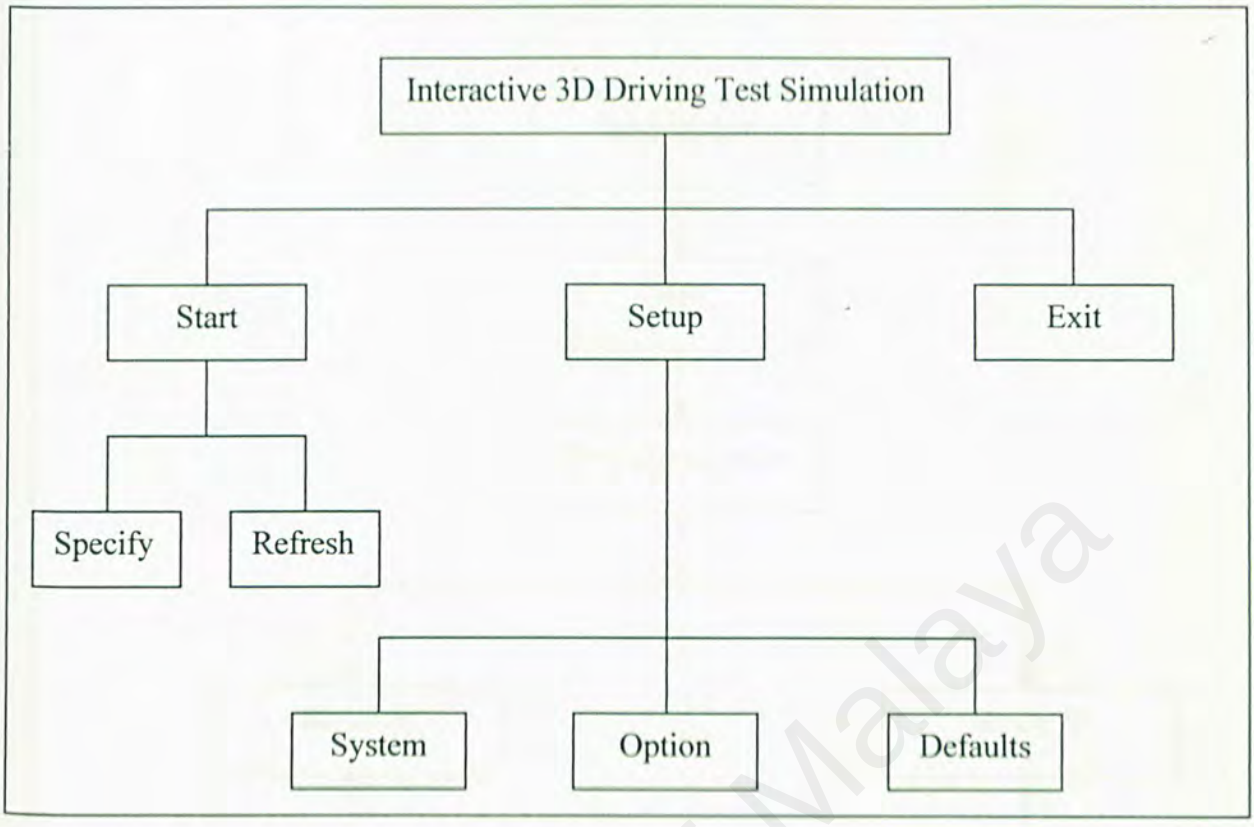


Figure 5.43: User Interface Engine

5.42 STORYBOARD INTERFACES

A storyboard interfaces is a graphical depiction of the outward appearance of the system, without any accompanying system functionality. A storyboard does not require much in terms of computing power to construct. In fact they can be mocked up without the aid of any computing resource (Dix et al, 1998). It is originally used in video production, and is now commonly used in multimedia projects (England and Finney, 1999). A storyboard is organized sequentially, screen by screen, and each screen is sketched out with design notes and specifications before rendering (Vaughan, 1999).

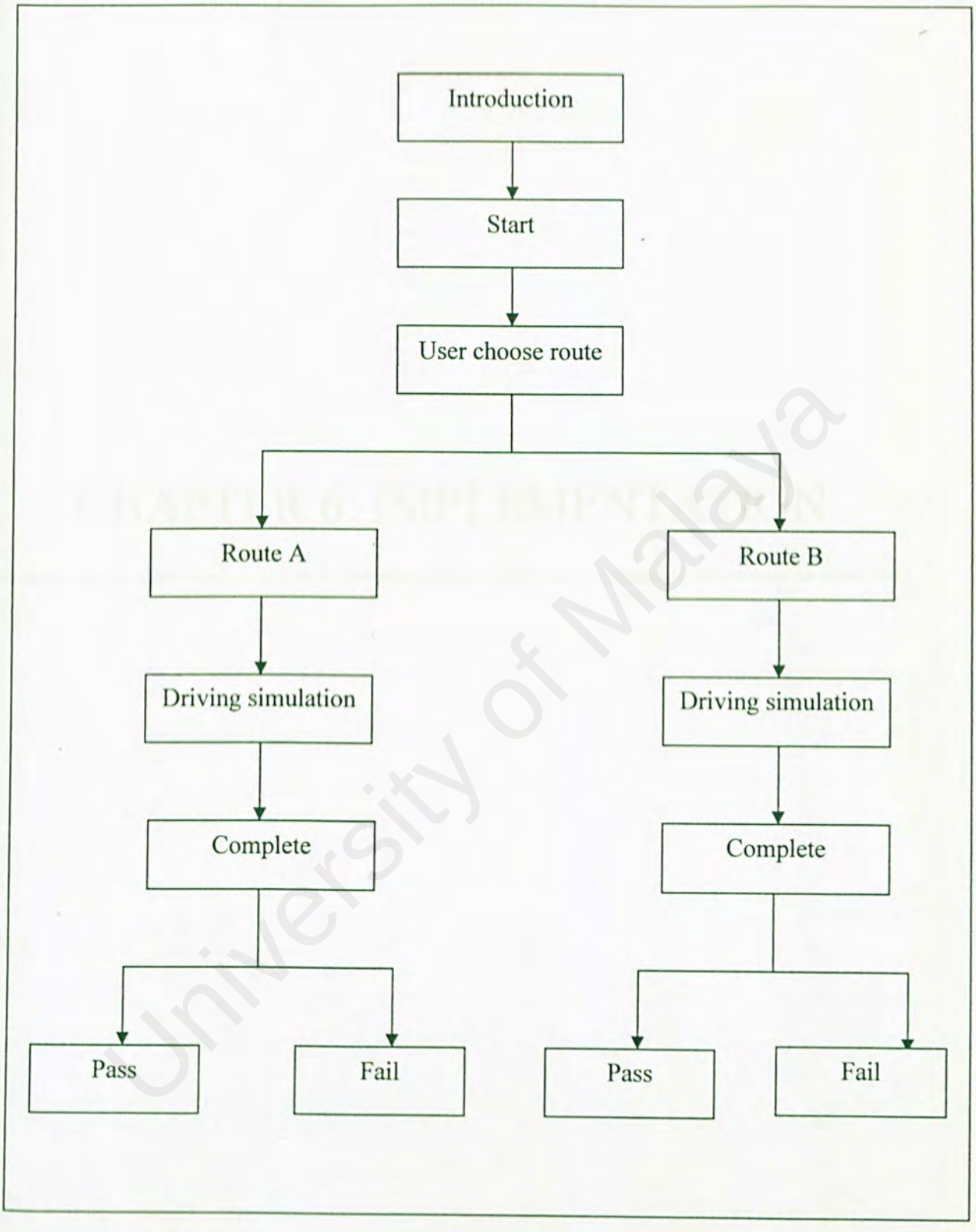


Figure 5.44: Storyboard of Interactive 3D Driving Test Simulation

CHAPTER 6

SYSTEM IMPLEMENTATION

INTODUCTION

After designing the proposed Interactive 3D Driving Test Simulation which specialized in 3D environment driving test, system implementation is then to be carried out. In order to develop system, installation and configuration of 3D GameStudio is first to be done, followed by system model and lastly generating source code. Coding and debugging are the major tasks involved in the implementation phase. This phase also will involve some modifications to the previous proposed design.

This chapter discussed the steps of methods taken to develop the system that was designed in the previous chapter.

System implementation is a process that was developed according to the system requirement and will be designed into program codes.

6.1 DEVELOPMENT ENVIRONMENT

Before the development a system, it is a must for the development to ensure that the software or hardware which will be used in the process of the development which will be used in the used in the process of development the system is really available. The hardware and software tools used to develop and document the entire Interactive 3D Driving Test Simulation is discussed as below.

6.11 HARDWARE CONFIGURATION

The development tools include hardware requirement for developers and users.

Main hardware for developers (minimum requirements):

- 3D Video Card or equivalent (64MB GeForce 4)
- Mouse
- 20GB hard disk
- Pentium 200MHz Processor or equivalent
- 128 MB Random Access Memory (RAM)
- CD-ROM drive
- CD-RW drive
- Sound Card
- Speakers
- DirectX 7.0 plug-in

User's main system requirements:

- Intel Pentium 200MHz Processor or equivalent
- Windows 98/NT/2000/XP
- 64 MB Random Access Memory (RAM)
- CD-ROM drive
- 640 x 480, 256 color display
- Sound Card
- Speakers
- Keyboard
- Mouse

6.1.2 SOFTWARE CONFIGURATION

A few of software tools are used in design the structural of the system and writing the report for the documentation purpose. The design process involves the drawing of the structured chart and other form of foundation of the software development. A vast array of software tools are listed as below:

software	**usage**	**description**
Microsoft Windows 2000 Professional	System requirement	Operating system
Adobe Photoshop 6.0	Interfaces design	Graphic design Authoring tools
Visual Studio	System development	System coding
3D GameStudio	System development	3D graphic design Model Editor Level Editor
Microsoft office 2000		System documentation

Top-Down Approach

System development is the process of creating program that is needed to satisfy the requirement of the system. There are two types of approaches in developing the system process, *top-down* and *bottom-up*.

The top down is a way of building the high level modules skeletons that are used to call the lower level modules that contains of function and procedures, where as the bottom-up approach is the reverse.

This system was build according to the top-down approach. The advantages of using top-down approach are:

- Start with high-level system and integrate from the top-down replacing individual components by stubs where appropriate.
- The top level is tested by itself
- All components called by the tested components are combined and tested as a larger unit
- This approach is reapplied until all components are incorporated
- Critical function can be prioritized to test their efficiency.

6.2 SYSTEM DEVELOPMENT OF INTERACTIVE 3D DRIVING TEST SIMULATION

This section explained the development of Interactive 3D Driving Test Simulation which focuses on the analysis of the usage of the technology and development tools that had been used. The system will be evolved from stretch design to a run able application. Below are the development steps carried our during the system development.

6.21 INTERFACE DEVELOPMENT

Attractive interface is important to attract users. The interface of Interactive 3D Driving Test Simulation is created using Adobe Photoshop and the menu interface is also created with Adobe and applies to the system using script that has been provided in 3D GameStudio software. The menu interface is including play, help, control and quit button.

The background of the menu interface is using black color because users will just focus to the play, control, help and quit button to choose which button they want to view.

6.211 Graphic Development

Adobe Photoshop 7.0 is the main graphic tool that is used to create or edit graphic. The dashboard, some of the sign board and e few more images in the project are created with the tool. Adobe Photoshop, everything is editable and unused graphics can be modifying to be a useful graphic. For this project, graphics are modified and exported to be a .mdl and .pcx format in order to reduce the size of the graphic.

6.212 3D GameStudio

3D GameStudio is the leading authoring system for all sorts of 2D and 3D realtime applications. It combines a high-end 3D engine, a 2D engine, a physics engine, a map and model editor, a script compiler and huge libraries of 3D objects, artwork and pre-made games. It was never easier to create 1st person games, 3rd person games, role playing games, side scrollers, flight simulators, board games, sports games, real-time presentations, virtual exhibitions and other 3D applications.

Three levels of games and other application creation:

- Games in the level editor - for simple games and other application
- Program games in C-Script - for ambitious commercial projects
- Program games in C++ - for professional developers

Games authoring systems are the future of game programming. Even with no programming knowledge, by following the step-by-step tutorial a simple action game can be built. Using the scripting language, games and other application in commercial quality can be created, successfully published and distributed.

The Editors

GameStudio contains a professional level, model and terrain editor for avoiding all the hassle with third party editors and format conversion programs. Creating landscapes, placing light effects, defining movement paths and modeling actors is a straightforward process. For quick results a huge library of over 1200 textures, building parts, furniture, vehicles, weapons and animated actors is included. Due to the import compatibility to most 3D formats and popular 3D games, levels, textures and 3D models can easily be created or downloaded from on the Internet.

Each level consists of an environment with fixed and moving parts - landscapes, dungeons, buildings or cities. Within the environment you can place an unlimited number of colored light sources, which throw real shadows, and 3D sound sources, which emit background sounds. Textures and Sprites can be created with any paint program. Animated 3D models are created with the model editor MED.

Developers can import levels and animated models created with popular 3D game editors like WorldcraftT or MilkshapeT. Usual 3D file formats (X, 3DS, MDL, MD2, MAP, WAD) are supported. Terrain can be imported from freeware terrain generators like TerragenT. If high-end editors like MayaT or 3D Studio MAXT are used, animated models with the MAYA or MAX plugin can be created. Through the MilkshapeT compatible filter interface, the model format of almost any game can be imported or exported.

As soon as the levels are finished, a button click generates the CD content ready for distribution. The optional File Packer compresses the files belonging to the game into a tiny encrypted resource.

6.213 Application Development using c-script

The C-Script Programming Language

GameStudio's C-Script language allows game programming from a beginner's to a professional level. This application allow developers to do something more ambitious like programming their own gameplay, player behaviour, effects, artificial intelligence, and user interface, instead of using the predefined scripts.

C-Script is a simplified version of C++, the language used for professional programming. C-Script is the best way to learn 'real' programming while entering code, developers can observe the results immediately in the virtual world.

Despite it's easy to learn, it's a real programming language with multitasking, arrays, structs, pointers, global and local variables, string and vector functions, file I/O and much more. You have a syntax highlighting editor, a debugger, and a compiler. Unlike Basic interpreters, C-Script is compiled .A C-Script program runs almost as fast as if it were 'directly programmed' into the engine. And speed is of the essence for most games and other application.

GameStudio's engine can also be driven by external DLLs through the plug-in interface developers prefer to use your own C++ or Delphi development system.

6.3 CODING TECHNIQUES

Coding is the process of translating the design specification into source codes that the computer can process. The source codes may require an interpreter or compiler and linker to convert them into an executable program. Without those coding, the system cannot worked successfully and precisely.

The program codes written must have appropriate comments, consistent and integrity of the terms used. This would make codes very easy to trace, debug and maintain by another programmer or developer. The purpose for coding includes:

- Classifying information
- Concealing information
- Revealing information
- Requesting information

6.31 Coding Methodology

The top-down approach was used as the coding methodology when developing the interactive 3D Driving Test Simulation System. A top-down approach is easy to visualize at it looks at the overall picture of the system and subsequently exploded into smaller and more manageable subsystems.

The top-down approach provides desirables emphasize on the synergy of the interface that systems and subsystems required. Furthermore, it ensures that the most important modules are developed and test first. Deploying the method gives preliminary version of the system sooner.

```
////////////////////////////////////////////////////////////////////////
// A5 MAIN WDL
////////////////////////////////////////////////////////////////////////
// FILES TO OVER-RIDE:
// * LOGODARK.BMP - THE ENGINE LOGO, INCLUDE YOUR GAME TITLE
// * HORIZON.PCX - A HORIZON MAP DISPLAYED OVER THE SKY AND
CLOUD MAPS
////////////////////////////////////////////////////////////////////////
// THE PATH KEYWORD GIVES DIRECTORIES WHERE GAME FILES CAN
BE FOUND,
// RELATIVE TO THE LEVEL DIRECTORY
PATH "C:\\PROGRAM FILES\\GSTUDIO\\TEMPLATE";      // PATH TO
WDL TEMPLATES SUBDIRECTORY
PATH "MODEL";
PATH "SOUND";
PATH "GAMBAR";
PATH "WAD";

////////////////////////////////////////////////////////////////////////
// THE INCLUDE KEYWORD CAN BE USED TO INCLUDE FURTHER WDL
FILES,
// LIKE THOSE IN THE TEMPLATE SUBDIRECTORY, WITH
PREFABRICATED ACTIONS
```

```
INCLUDE <MOVEMENT.WDL>;
INCLUDE <MESSAGES.WDL>;
INCLUDE <MENU.WDL>;          // MUST BE INSERTED BEFORE DOORS
AND WEAPONS
INCLUDE <PARTICLE.WDL>; // REMOVE WHEN YOU NEED NO PARTICLES
INCLUDE <DOORS.WDL>;         // REMOVE WHEN YOU NEED NO DOORS
INCLUDE <ACTORS.WDL>;  // REMOVE WHEN YOU NEED NO ACTORS
INCLUDE <WEAPONS.WDL>; // REMOVE WHEN YOU NEED NO WEAPONS
INCLUDE <WAR.WDL>;    // REMOVE WHEN YOU NEED NO FIGHTING
//INCLUDE <VENTURE.WDL>;  //   INCLUDE   WHEN   DOING   AN
ADVENTURE
INCLUDE  <LFLARE.WDL>;    // REMOVE WHEN YOU NEED NO LENS
FLARES
INCLUDE <MOVE_CAR.WDL>;
INCLUDE <MENU1.WDL>;
INCLUDE <USERCTRL.WDL>;
INCLUDE <HELP.WDL>;
INCLUDE <CONTROLMENU.WDL>;

//////////////////////////////////////////////////////////////////
// THE ENGINE STARTS IN THE RESOLUTION GIVEN BY THE FOLLWING
VARS.
VAR VIDEO_MODE = 6;    // SCREEN SIZE 640X480
```

```
VAR VIDEO_DEPTH = 16; // 16 BIT COLOUR D3D MODE

//////////////////////////////////////////////////////////
// STRINGS AND FILENAMES
// CHANGE THIS STRING TO YOUR OWN STARTING MISSION MESSAGE.
STRING MISSION_STR = "INSTRUCTION ";
STRING MISSION1_STR = "1. THE MAIN MENU CONSIST OF PLAY, CONTROL, HELP AND QUIT BUTTON. ";
STRING MISSION2_STR = "2. CHOOSE PLAY BUTTON TO START. ";
STRING MISSION3_STR = "3. CHOOSE CONTROL BUTTON TO VIEW THE CONTROL KEY. ";
STRING MISSION4_STR = "4. CHOOSE HELP BUTTON TO VIEW HELP MENU. ";
STRING LEVEL_STR = <MAIN.WMB>; // GIVE FILE NAMES IN ANGULAR BRACKETS

//////////////////////////////////////////////////////////
// DEFINE A SPLASH SCREEN WITH THE REQUIRED A4/A5 LOGO
BMAP SPLASHMAP = <INTRO.PCX>; // THE DEFAULT LOGO IN TEMPLATES
PANEL SPLASHSCREEN {
        BMAP = SPLASHMAP;
        FLAGS = REFRESH,D3D;
```

```
}

//////////////////////////////////////////////////
// SCREEN TEXT FOR STEERING MODE - KEYBOARD OR JOYSTICK
TEXT STRINGINS
 {
  POS_X = 15;
  POS_Y = 10;
  LAYER = 3;
  FONT = PANEL_FONT;
  STRING = MISSION_STR;
//   FLAGS = TRANSPARENT;
 }


//////////////////////////////////////////////////
// SCREEN TEXT FOR STEERING MODE - KEYBOARD OR JOYSTICK
TEXT STRING1
 {
  POS_X = 15;
  POS_Y = 30;
  LAYER = 3;
  FONT = PANEL_FONT;
```

```
   STRING = MISSION1_STR;
//   FLAGS = TRANSPARENT;
  }

////////////////////////////////////////////////
// SCREEN TEXT FOR STEERING MODE - KEYBOARD OR JOYSTICK
TEXT STRING2
 {
  POS_X = 15;
  POS_Y = 50;
  LAYER = 3;
  FONT = PANEL_FONT;
  STRING = MISSION2_STR;
//   FLAGS = TRANSPARENT;
 }

////////////////////////////////////////////////
// SCREEN TEXT FOR STEERING MODE - KEYBOARD OR JOYSTICK
TEXT STRING3
 {
  POS_X = 15;
  POS_Y = 70;
  LAYER = 3;
```

```
   FONT = PANEL_FONT;
   STRING = MISSION3_STR;
//   FLAGS = TRANSPARENT;
  }

////////////////////////////////////////////////
// SCREEN TEXT FOR STEERING MODE - KEYBOARD OR JOYSTICK
TEXT STRING4
  {
   POS_X = 15;
   POS_Y = 90;
   LAYER = 3;
   FONT = PANEL_FONT;
   STRING = MISSION4_STR;
//   FLAGS = TRANSPARENT;
  }

////////////////////////////////////////////////////////////
// THE FOLLOWING SCRIPT CONTROLS THE SKY MOVEMENT
BMAP HORIZON_MAP = <NEWHORIZON.PCX>;
// A BACKDROP TEXTURE'S HORIZONTAL SIZE MUST BE A POWER OF 2;
// THE VERTICAL SIZE DOES NOT MATTER
```

```
FUNCTION INIT_ENVIRONMENT()
{
        SCENE_MAP = HORIZON_MAP;  // HORIZON BACKDROP OVERLAY
        SCENE_NOFILTER = ON;

        SKY_SPEED.X = 1;
        SKY_SPEED.Y = 1.5;
        CLOUD_SPEED.X = 3;
        CLOUD_SPEED.Y = 4.5;

        SKY_SCALE = 0.5;
        SKY_CURVE = 1;                    // 'STEEPNESS' OF SKY DOME

        SCENE_FIELD = 60;                 // REPEAT MAP 6 TIMES
        SCENE_ANGLE.TILT = -10; // LOWER EDGE OF SCENE_MAP 10
UNITS BELOW HORIZON

        SKY_CLIP = SCENE_ANGLE.TILT;      // CLIP THE SKY AT BOTTOM
OF SCENE_MAP
}

FUNCTION INTRO_HELP_ON()
{
```

```
        STRINGINS.VISIBLE = ON;
        STRING1.VISIBLE = ON;
        STRING2.VISIBLE = ON;
        STRING3.VISIBLE = ON;
        STRING4.VISIBLE = ON;
}

FUNCTION INTRO_HELP_OFF()
{
        STRINGINS.VISIBLE = OFF;
        STRING1.VISIBLE = OFF;
        STRING2.VISIBLE = OFF;
        STRING3.VISIBLE = OFF;
        STRING4.VISIBLE = OFF;
}

////////////////////////////////////////////////////////
// THE MAIN() FUNCTION IS STARTED AT GAME START
FUNCTION MAIN()
{

        //INIT_CONTROLS();
```

```
// SET SOME COMMON FLAGS AND VARIABLES
//      WARN_LEVEL = 2; // ANNOUNCE BAD TEXTURE SIZES AND BAD
WDL CODE
        TEX_SHARE = ON;  // MAP ENTITIES SHARE THEIR TEXTURES

// CENTER THE SPLASH SCREEN FOR NON-640X480 RESOLUTIONS, AND
DISPLAY IT
        SPLASHSCREEN.POS_X      =      (SCREEN_SIZE.X      -
BMAP_WIDTH(SPLASHMAP))/2;
        SPLASHSCREEN.POS_Y      =      (SCREEN_SIZE.Y      -
BMAP_HEIGHT(SPLASHMAP))/2;
        SPLASHSCREEN.VISIBLE = ON;
// WAIT 3 FRAMES (FOR TRIPLE BUFFERING) UNTIL IT IS FLIPPED TO THE
FOREGROUND
        WAIT(3);

// INIT THE 'ENVIRONMENT' (SKY & SCEEN MAP)
        INIT_ENVIRONMENT();
// NOW LOAD THE LEVEL
        LEVEL_LOAD(LEVEL_STR);
// FREEZE THE GAME
        FREEZE_MODE = 1;
```

```
// WAIT THE REQUIRED SECOND, THEN SWITCH THE SPLASHSCREEN
OFF.
        WAITT(100);
        SPLASHSCREEN.VISIBLE = OFF;
        BMAP_PURGE(SPLASHMAP);     // REMOVE SPLASHSCREEN FROM
VIDEO MEMORY

// LOAD SOME GLOBAL VARIABLES, LIKE SOUND VOLUME
        LOAD_STATUS();

// DISPLAY THE INITIAL MESSAGE
        //MSG_SHOW(MISSION_STR,100);
        INTRO_HELP_ON();
        WAITT(200);
        INTRO_HELP_OFF();

// INITIALIZE LENS FLARES WHEN EDITION SUPPORTS FLARES
IFDEF CAPS_FLARE;
        LENSFLARE_START();
ENDIF;

// UN-FREEZE THE GAME
        FREEZE_MODE = 0;
```

```
//      CLIENT_MOVE();   // FOR A POSSIBLE MULTIPLAYER GAME
// CALL FURTHER FUNCTIONS HERE.....

        AFFICHE_MENU();
}

////////////////////////////////////////////////////////////
// THE FOLLOWING DEFINITIONS ARE FOR THE PRO EDITION WINDOW
COMPOSER
// TO DEFINE THE START AND EXIT WINDOW OF THE APPLICATION.
WINDOW WINSTART
{
        TITLE           "3D GAMESTUDIO";
        SIZE            480,320;
        MODE            IMAGE;      //STANDARD;
        BG_COLOR        RGB(240,240,240);
        FRAME                   FTYP1,0,0,480,320;
//      BUTTON
        BUTTON_START,SYS_DEFAULT,"START",400,288,72,24;
        BUTTON
        BUTTON_QUIT,SYS_DEFAULT,"ABORT",400,288,72,24;
```

```
        TEXT_STDOUT     "ARIAL",RGB(0,0,0),10,10,460,280;
}

/* NO EXIT WINDOW AT ALL..
WINDOW WINEND
{
        TITLE           "FINISHED";
        SIZE            540,320;
        MODE            STANDARD;
        BG_COLOR        RGB(0,0,0);
        TEXT_STDOUT     "",RGB(255,40,40),10,20,520,270;

        SET FONT        "",RGB(0,255,255);
        TEXT            "ANY KEY TO EXIT",10,270;
}*/

////////////////////////////////////////////////////////
//INCLUDE <DEBUG.WDL>;

//ON_F1=INTRO_HELP_ON;
```

Main program for the system: This script contains the variables, subprogram and declaration to compile and run the system.

6.4 CHAPTER SUMMARY

This chapter explains the details of the system development procedures various kinds of software and hardware were used to develop this system. The coding methodology used was top-down methodology. The main three modules were developed using variety of tools in different kinds of techniques. Microsoft script debugger has helped out in discovering the error.

CHAPTER 7

SYSTEM TESTING

Testing is done from time to time to the program from the day the coding begun. The purpose of testing is to ensure the resulting component of program as well as the program as a while fulfils the requirement specification and to eliminate faults in the program. Due to the errors that has been done during the system development or system design, faults, and failures may happen even when the entire system has been developed. Therefore, the main idea of testing is to demonstrate correctness of the program, identify the errors in the system coding or the system design. The faults that are discovered during the testing procedures must be corrected before launching the system.

7.1 TESTING STRATEGIES

Testing Strategy is a strategy of establishing the existence of errors. It is also a general approach to the testing process rather than a method of devising particular system or component tests.

The testing strategies consist of following method:

- ***Top-down testing***

  Testing starts with the most abstract component and works downwards until all the modules are tested.

- ***Bottom-up testing***

  It is one of the popular approaches used to test large systems. Testing starts with the fundamental components and works upwards.

- ***Back-to-back testing***

  Used when versions of a system are available. The systems are tested together and their outputs are compared.

- ***Thread testing***

  Used for system with multiple processes where the processing of transition threads its way through these processes.

7.2 SYSTEM TESTING

System testing can be divided into unit testing, integration testing and system testing. Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. A good test case has a high probability of finding an undiscovered error from the system.

The main objectives of system testing are:

- To reveal different classes of errors and do so with a minimum mount of time and effort.
- To ensure the function appear to be working according to the specification.
- To demonstrates that behavioral and performance appear to have been met.

It is important to let users to test on the system to give a view of the system and any comment from the user is useful to enhance this system based on their requirement.

For this system, unit or module testing and top-down testing have been carried out for integration testing.

7.21 Unit Testing

Unit testing is to test the program in each module independently. The primary goal of unit or module testing is to confirm that the unit is correctly coded and that it carries function it is suppose to carry out. It is the initial testing stage for the completion of each component class. The interaction testing between components are initially avoided and to carried out later in the bottom-up integration testing.

Unit testing manages the combination of testing, it facilitates errors diagnose and correction by development and it allows parallelism, in other words it test multiple function. All the logical error that contain in the classes will be detected. The following areas were tested during unit testing for this project:

- Boundary values analysis.

  Ensure that the module operates properly at boundaries established to limited or restricted processing.

- Handling Paths

  Ensure that the specific module executes the recovery process while an error occurs.

- All possible independent program paths are executed.

  Ensure that the control structures are implemented correctly.

Examples of test cases

- Ensure that the click on button will function appropriately and only click once when it is click.
- To ensure the module will function as it is specified.

Unit testing helps to correct the error an all the module. A few errors were identified and corrected after carrying out unit testing.

7.22 Integration Testing

Integrated testing is a systematic technique for constructing the program structure while at the same time conducting test to discover the errors. Integration testing is used to ensure that the system will work correctly when all the models and script files combined together. Each model is tested individually first to see the resultant of combining together with the scripts. Then it will be tested to whole system to completing the system. This approach will repeat until all the models being tested.

The goal of carrying out integration testing is to take unit tested components and build a program structure that has been dictating by design. This testing will ensure that each module is arranged and functioning correctly.

7.23 Function Testing

Function testing focus on the functionality of the system. It is based on the system functional requirement and it checks that the integration on the system perform it functions as specified on the requirement.

7.24 Performance Testing

This testing usually involved the hardware as well as software, when the system performs the function required by the requirements, the testing process then turn to test the way in which those functions are performed. Thus the performance testing addresses the non-functional requirements. The purpose of this testing is to test the run time performance of this software within the context of an integrated system.

7.25 Acceptance Testing

This is one of the stages in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system producer rather than simulated test data. Sometimes, this testing is called as alpha testing and this testing process continue until the system developer and client agree that the delivered system is an acceptance implementation of the system requirement.

7.3 CHAPTER SUMMARY

Testing is the most important steps in developing the system perfectly. Precision and accuracy of output data or appearance is considered during this process. Unit, integration and system testing has been carried out for this systems. The objective of a system will only achieve after all the through testing done by different user with different aspects.

This chapter describes the types of testing done on the system. There were few types of testing did on the system. All types testing were done to make sure no errors in the complete system before handling it over to the client. The last testing was the acceptance testing tested by the client.

CHAPTER 8

SYSTEM EVALUATION

The development and implementation of the I3DDTS proved to be a challenging task as problems arise along the way. The basic knowledge needed as foundation in building an application of this nature involves deep studies in certain field such as the 3D development tool, programming language and user interface development tool.

In this phase, this system was evaluated to identify its strengths, limitations, and proposals were written for the future enhancements.

8.1 PROBLEMS ENCOUNTERED AND SOLUTIONS

There are several problems encountered in the process of developing I3DDTS. The problem faces are as below:

8.11 Difficulties In Determining The Scope Of The System

It is impossible to build a full-scale complete real-time Driving test route in this system within the short time frame and a limited knowledge in 3D development tool. Not all the particular buildings, road and object such as trees, sign board, car etc are put in the system. Advices and opinions were given by project supervisor to outline the scope of the project to build during initial stages. Furthermore, the results of survey and interview have also contributed for an outlook of the system scope.

8.12 Difficulty In Designing Telepresence Environments

With lack experience in designing an interactive environments, standard and user-friendly interface, makes designing phase become more difficult and challenging. To gain more information on the designing phase, lots of interesting web sites were visited and deeply studied. One of the most important web site was www.3dgamestudio.com which has become major source in developing this system.

Besides that, "try and error" method were the most implemented method in developing proper scripts and standard 3d's design.

8.13 Difficulty In Rendering The System

The software however needs a high RAM (Read Access Memory) to render the environments in the system. It takes about 64 RAMs to render only 5000 portals; that is a small number of portal compare with what the system need.

8.2 SYSTEM STRENGTHS

There are several advantages of this system as listed below:

8.21 User Friendliness

The interface of the system is user friendly and easy to navigate and understand. Besides, this system being included with manual; within hardcopy or through the system help menu.

8.22 Easy To Use

This system is very easy to use. The commands and the layouts are simple and well organized. Therefore, it is easy to learn up, use, navigate and remember a certain road in the driving test area.

8.23 Easy Access

This system is offline software and can be download or install in a PC. It can run on almost all PC in the world right now; either MAC or Windows.

8.24 Real-Time Rendering

This system can be considered fast enough if we compare with other 3d's player; such like VRML or OpenGL. System interface have been in real-time rendering, where its become more realistic than others authoring tools. The movements of the system are smooth; without have to render again the environments.

8.25 Free From 3rd Party Platform

Unlike other software developments tools, these systems are stand-alone player without have to download other's platform for to run the system. Thus, its makes the user just 'click and run' the programs and it have been prove that so conveniences to the user.

8.3 SYSTEM LIMITATIONS

However, there are limitations in these systems that are not resolved yet.

8.31 Lack Of Detail Structure

The system could not include a lot of structure due to PC limitation on designing phase. Detail structure in a system mainly could not being build with ordinary PCs like Windows, due to lack of memory usability and performance of the PC's processor.

8.32 Limited Functionality

This system only provides through offline usage. User can only download and install the program through their PCs. It could not run as web-base application due to the bandwidth of Internet connection and the associated files that needed the system to runs smoothly. Therefore, the system cans only runs by user's PCs.

8.4 FUTURE ENHANCEMENT

The system should be maintained throughout the lifetime of the system because the user requirements might vary from times to times. Enhancement in the future will extend the usability of this system. Moreover, the system limitations should be improved to enhance functionality.

Here are some suggestion and possible future enhancements:

8.41 Enhance User Interface

User interface should enhance from time to time. Multimedia elements such as streaming video, more graphics especially animated graphics or animated entities should be added to increase its attractiveness, impressive and interactive and to make the system multimedia map program.

8.42 More Functionality Added

Dynamic key configuration should be added to make sure the system more users friendly and flexible for user to change the configuration.

8.43 Develop For Other Platform

The system should be design web-base to make sure that user can use the system globally.

8.44 Language Flexibility

Since the usage of this system will be used world wide, other language should be implemented to the system for different user's language

8.5 CHAPTER SUMMARY

This chapter evaluates Interactive 3D Driving Test Simulation fron different perspectives. It's begins by stating the problems encountered throughout the development and implementation of the system which includes the challenges in learning and development tools. Several problems were encountered during the development phase of this system. However it was solved with the helps of my partner and friends and supervisor and lecturers.

Developing this system has given me the chance to learn variety kind of programming language and tools. It has also exposed me different minds of latest technology in computer industry.

This system can be enhanced to a better stand alone or web based system in different kinds of aspects in the future.

REFERENCES

[1] Royal Malaysian police (2003). *Statistik Kemalangan Jalan Raya* [online]. Available from: http://www.rmp.gov.my/pdrm02/pdrm.htm [Accessed 28 June 2003]

[2] Oxford Paperback Reference (1996). *A Dictionary of Computing 4th ed.* Oxford New York: Oxford University Press.

[3] Xiaopeng Fang, Hung A. Pham and Carolina, Cruz-Neira (2002). *Driving In the Virtual World* [online]. Department of Mechanical Engineering, Iowa State University. Available from: http://www.vrac.iastate.edu/~fang414/Publications/Paper_Driving.doc [Accessed 20 July 2003]

[4] *Virtual Driving Simulator* [online]. Available from: http://www.teendrivingcourse.com/DVDTraining/3dillustrations.cfm [Accessed 01 September 2003]

[5] *Raydon Virtual Driver: Driving Simulators* [online]. Available from: http://www.virtualdriver.net [Accessed 30 August 2003]

[6] J. Kearney (2002). *Behavior Modeling and Scenario Authoring for Virtual Environments* [online].Available from: http://www.cs.uiowa.edu/~kearney/scenario.html [Accessed 12 August 2003]

[7] Martin Baker (2003). *Programming Language for 3D simulation and games* [online]. Available from: http://www.martinb.com/software/language/ [Accessed 6 August 2003]

[8] Martin Baker (2003). *Programming Language for 3D simulation and games* [online]. Available from: http://www.euclideanspace.com/threed/rendering/opengl/index.htm [Accessed 6 August 2003]

[9] Refsnes Data (2003). *Browser Statistics* [online] http://www.w3schools.com/browsers/browsers_stats.asp [Accessed 16 June 2003]

[10] Magdy F. Iskander, J. Corey Catten, Antony Jones, Rex Jameson, and Albert Balcells (2000). *Interactive Multimedia Lessons for Education*. CAEME Center, College of Engineering, University of Utah.

[11] P.A Egerton and W.S Hall (1998). *Computer Graphic*. Europe: Prentice Hall.

[12] Jamaluddin Harun and Zaidatun Tasir (2000). *Pengenalan Kepada Multimedia*. Kuala Lumpur: Venton Publishing.

[13] WordNet Dictionary (2003). *Driving* [online]. Available from: http://www.hyperdictionary.com/dictionary/driving [Accessed 7 September 29, 2003]

[14] whatis.com (2003). *C++ Definition* [online]. Available from: http://searchdatabase.techtarget.com/sDefinition/0,,sid13_gci211850,00.html [Accessed 15 August 2003]

[15] Russell Kay (2003) *Quick Study: System Development Life Cycle* [online]. Available from: http://www.computerworld.com/developmenttopics/development/story/0,10801,71151,00.html [Accessed 18 August 2003]

[16] whatis.com (Jul 13, 2001) *waterfall model* [online]. Available from: http://searchvb.techtarget.com/gDefinition/0,294236,sid8_gci519580,00.html [Accessed 18 August 2003]

[17] Marilyn Meyer and Roberta Baber (1998) *Computers Today and Tomorrow* [online]. Available from: http://www.mis.boun.edu.tr/ulus/ibs511/notlar/analysis.htm [Accessed 28 August 2003]

[18] *Introduction to VRML* [online]. Available from: http://www.virtualrealms.com.au/vrml/tute01/intro_to_vrml.htm [Accessed 23 August 2003]

[19] Scherer, D., Dubois, P. and Sherwood, B. (2000). *VPython: 3D Interactive Scientific Graphics for Students,* Computing in Science and Engineering, Sept /Oct. 2000, 82-88.

[20] Angel, Edward (2003). *Interactive Computer Graphics: A Top-Down approach with OpenGL 3rd ed.* United State of America: Addison Wesley.

[20] Jeffrey L Whitten, Lonnie D. Bentley and Kevin C. Dittman (2002). *System Analysis and Design Methods 5th ed.* Singapore: McGraw Hill.

[21] Shari Lawrence Pfleeger (2001). *Software Engineering: Theory and Practice 2nd ed.* United State of America: Prentice Hall International Editions.

[22] Ian Sommerville (2001). *Software Engineering 6th ed.* United State of America: Addison Wesley.

[23] Bjarne Stroustrup (2003). *The C++ Programming Language* [online]. http://www.research.att.com/~bs/C++.html [Accessed 20 August 2003]

[24] Dennis M. Ritchie (2003). *The Development of the C Language* [online]. Available from: http://cm.bell-labs.com/cm/cs/who/dmr/chist.html [Accessed 25 August 2003]

[25] *Screenshots Gallery* [online]. Available from: http://www.conitec.net/gallery.htm [Accessed 31 August 2003]

[26] Jarrell Pair (2001). *Virtual Environments: Challenges and Opportunities* [online]. Digital Worlds Institute University of Florida. Available from: http://www.digitalworlds.ufl.edu/resources/presentations/ve.pdf [Accessed 04 September 2003]

[27] International New Students' Edition (1995). *Oxford: Advanced Learner's Dictionary.* New York: Oxford University Press

[28] Kenneth E. Kendall and Julie E. Kendall (1995). *System Analysis and Design 3rd ed.* United State of America: Prentice Hall International, Inc

[29] (September 1997). *A new type of driver's aid for road simulation* [online]. Available from: http://www.cbb-software.com/products/flg.html [Accessed 19 August 2003]

[30] About.com (2003). *Simulation* [online]. Available from: http://3dgraphics.about.com/library/glossary/bldef-simulation.htm [Accessed 5 September 2003]

[31] U.S. Department of Transportation National Highway Traffic Safety Administration (December 1996) [online]. Available from: http://www.noviceforum.drivers.com/papers/noviceforum2.pdf [Accessed 5 August 2003]

[33] opengl.org (2003). *What is OpenGL, why you need it and where you get it* [online]. Available from: http://www.opengl.org/users/about/index.html [Accessed 15 September 2003]

[34] GCN (October 12, 1998). *Sate visual appetite with Studio Max* [online]. Available from: http://www.gcn.com/archives/gcn/1998/October12/28.htm [Accessed 18 September 2003]

[35] Alias.com (2002-2003). *The Maya Family* [online]. Available from: http://www.alias.com/eng/products-services/maya/index.shtml [Accessed 28 September 2003]

[36] Microsoft.com (2003). *Technology Overview* [online]. Available from: http://www.microsoft.com/directx [Accessed 15 September 2003]

[37] Lecture Notes (2003). *Models in Software Development* [online]. Available from: http://perdana.fsktm.um.edu.my/~wxes3204/SE_1_2003_4/sw_models.ppt [Accessed 24 August 2003]